



Provincia de Río Negro
CONSEJO PROVINCIAL DE EDUCACIÓN

VIEDMA, 11 DE MARZO DE 2022

VISTO:

El Expediente N° 97249-EDU-2015 del Registro del Ministerio de Educación y DD HH - Consejo Provincial de Educación, y;

CONSIDERANDO:

Que mediante el mismo se tramita la aprobación de los Diseños Curriculares Jurisdiccionales de las Tecnicaturas Superiores;

Que la Ley Nacional de Educación N° 26206, en su Artículo 17°, establece que la estructura del Sistema Educativo Nacional comprende cuatro niveles - Educación Inicial, Educación Primaria, Educación Secundaria y Educación Superior y ocho modalidades, entre las cuales se incluye como tal a la Educación Técnico Profesional;

Que indica, además, que la Educación Técnico Profesional se rige por las disposiciones de la Ley de Educación Técnico Profesional N° 26058 y es la modalidad de la Educación Secundaria y la Educación Superior responsable de la formación de técnicos secundarios y técnicos superiores en áreas ocupacionales específicas y de la Formación Profesional;

Que la Ley Orgánica de Educación Provincial F N° 4819 establece en su Artículo 122, las funciones de los Institutos Técnicos Superiores comprendiéndose la formación técnicos superiores que se caractericen por la solidez de su perfil profesional, la actualización de sus conocimientos y por su compromiso con el desarrollo industrial y productivo de la provincia y del país;

Que por Resolución N° 13/07 del Consejo Federal de Educación, se aprueba el documento “Títulos y Certificados de la Educación Técnico Profesional”;

Que por Resolución N° 13/07 del Consejo Federal de Educación, se aprueba el Marco de Referencia de Desarrollo de Software – Anexo V;

Que por Resolución N° 229/14 del Consejo Federal de Educación se aprueba el documento “Criterios Federales para la organización Institucional y Lineamientos curriculares de la Educación Técnica Profesional de Nivel secundario y superior;



Provincia de Río Negro
CONSEJO PROVINCIAL DE EDUCACIÓN

Que la Resolución N° 295/16 del Consejo Federal de Educación aprueba el Documento “Criterios Federales para la Organización Institucional y Lineamientos Curriculares de la Educación Técnico Profesional de Nivel Secundario y Superior”;

Que corresponde dictar la norma legal que apruebe el Diseño Curricular Jurisdiccional de la Tecnicatura Superior en Desarrollo de Software Full Stack, y la denominación de título;

POR ELLO, y de acuerdo a las facultades conferidas por el Artículo 165° de la Ley F N° 4819

LA PRESIDENTA
DEL CONSEJO PROVINCIAL DE EDUCACION
R E S U E L V E:

ARTÍCULO 1°.- APROBAR el Diseño Curricular Jurisdiccional de **“Tecnicatura Superior En Desarrollo de Software Full Stack”**, que como Anexo I forma parte integrante de la presente Resolución.-

ARTÍCULO 2°.-ESTABLECER que el título a otorgar será: **“Técnico Superior en Desarrollo de Software Full Stack”**.-

ARTÍCULO 3°.-ESTABLECER que el presente Diseño Curricular Jurisdiccional se encuentra encuadrado en la normativa que emite el Consejo Federal de Educación y el Ministerio de Educación de la Nación, que regulan la Educación Técnico Profesional para la Educación Superior.-

ARTÍCULO 4°.-REGISTRAR, comunicar por la Secretaría General a la Dirección de Educación Superior y por su intermedio a los interesados y archivar.-

RESOLUCIÓN N° 1599
DES/SG/cau.-

Prof Mercedes Jara Tracchia
Presidente del Consejo Provincial de Educación
Provincia de Río Negro

Diseño Curricular Jurisdiccional Año 2022

TECNICATURA SUPERIOR EN DESARROLLO
DE SOFTWARE FULL STACK

AUTORIDADES PROVINCIALES DE RÍO NEGRO

GOBERNADORA

Arabela CARRERAS

MINISTRA DE EDUCACIÓN

María de las Mercedes JARA TRACCHIA

SECRETARIO DE EDUCACIÓN

Horacio Adrián CARRIZO

CONSEJO PROVINCIAL DE EDUCACIÓN

Vocales Gubernamentales CPE:

Omar Eduardo RIBODINO

María Angélica PORTALES

DIRECTORA GENERAL DE EDUCACIÓN

Laura RONCATI

DIRECTORA DE EDUCACIÓN SUPERIOR

Marisa HERNÁNDEZ

Año 2022

EQUIPO JURISDICCIONAL CURRICULAR

Myriam VAZQUEZ

Lorena LOPEZ FRANK

Julieta ALBRIEU

Jessica CONTRERA

EQUIPO INSTITUCIONAL

INSTITUTO TÉCNICO SUPERIOR CIPOLLETTI

Rector: Javier Alejandro GÓMEZ BIANCHI

Secretaria Académica: Gabriela Larissa CAPIZZANO

Docente: Desarrollador Sergio Alejandro ARRIAGADA

Docente: Lic. Exequiel Walter Fernando WIEDERMANN

Docente: SysAdmin Julián Sebastián ROSAS

Docente: SysAdmin Mauro Alejandro BENEGA

ÍNDICE

CAPÍTULO I. MARCO DE LA POLÍTICA EDUCATIVA PROVINCIAL Y NACIONAL PARA LA EDUCACIÓN TÉCNICO PROFESIONAL	Pág.5
1.1 La Educación Técnico Profesional en la Provincia de Río Negro. Antecedentes y nuevos contextos.	Pág.5
CAPÍTULO II. FINALIDADES DE LA FORMACIÓN TÉCNICA EN DESARROLLO DE SOFTWARE FULL STACK	Pág.7
2.1 La Tecnicatura Superior en Desarrollo de Software Full Stack, aproximaciones a su campo de estudio, conocimiento.	Pág.9
2.2 Descripción de la Carrera.	Pág.9
2.3 Identificación del Título.	Pág.9
2.4 Denominación del título.	Pág.9
2.5 Duración de la Carrera.	Pág.9
2.6 Carga horaria de la Carrera.	Pág.9
2.7 Objetivos de la Carrera.	Pág.9
2.8 Campo Ocupacional.	Pág.10
2.9 Perfil del Egresado.	Pág.10
2.10 Condiciones de ingreso	Pág.11
CAPÍTULO III. FUNDAMENTOS PEDAGÓGICOS DE LA PROPUESTA CURRICULAR.	Pág.12
3.1 Acerca del Currículum, el Conocimiento, la Enseñanza y el Aprendizaje	Pág.12
3.2 Consideraciones Metodológicas	Pág.13
3.3 Acerca de la Evaluación	Pág.14
CAPÍTULO IV. ORGANIZACIÓN CURRICULAR	Pág.16
4.1 Definición y caracterización de los campos de la formación y sus relaciones	Pág.16
4.2 Carga horaria por campo	Pág.16
4.3 Definición de los formatos curriculares que integran la propuesta	Pág.17
CAPÍTULO V. ESTRUCTURA CURRICULAR	Pág.19
5.1 Mapa curricular tecnicatura	Pág.19
CAPÍTULO VI. UNIDADES CURRICULARES	Pág.20
6.1 Presentación de las Unidades Curriculares. Componentes básicos	Pág.20
6.2 CAMPO DE LA FORMACIÓN GENERAL	Pág.20
6.2.1 PRIMER AÑO	Pág.20
6.2.1.1 Inglés Técnico I	Pág.20
6.2.2 SEGUNDO AÑO	Pág.21
6.2.2.1 Inglés Técnico II	Pág.21
6.2.3 TERCER AÑO	Pág.21
6.2.3.1 Ética y Responsabilidad Social	Pág.21

6.2.3.2	Inglés Técnico III	Pág.22
6.3	CAMPO DE LA FORMACIÓN DEL FUNDAMENTO	Pág.23
6.3.1	PRIMER AÑO	Pág.23
6.3.1.1	Matemática	Pág.23
6.3.1.2	Laboratorio Full-Stack I	Pág.24
6.3.1.3	Arquitectura de las Computadoras	Pág.24
6.3.2	SEGUNDO AÑO	Pág.25
6.3.2.1	Laboratorio Full-Stack II	Pág.25
6.3.2.2	Integración de Aplicaciones	Pág.26
6.3.3	TERCER AÑO	Pág.26
6.3.3.1	Innovación y Desarrollo Emprendedor	Pág.26
6.4	CAMPO DE LA FORMACIÓN ESPECÍFICA	Pág.27
6.4.1	PRIMER AÑO	Pág.27
6.4.1.1	Programación	Pág.27
6.4.1.2	Base de datos	Pág.28
6.4.1.3	Diseño UX/UI	Pág.29
6.4.2	SEGUNDO AÑO	Pág.30
6.4.2.1	Desarrollo de Software	Pág.30
6.4.2.2	Programación Backend	Pág.31
6.4.2.3	Programación Frontend	Pág.31
6.4.3	TERCER AÑO	Pág.32
6.4.3.1	Desarrollo web	Pág.32
6.4.3.2	Desarrollo móvil	Pág.33
6.4.3.3	Gestión de Proyectos de Software	Pág.34
6.5	CAMPO DE LA FORMACIÓN DE LA PRÁCTICA PROFESIONALIZANTE	Pág.34
6.5.1	PRIMER AÑO	Pág.36
6.5.1.1	Unidad curricular: Práctica Profesionalizante I	Pág.36
6.5.2	SEGUNDO AÑO	Pág.37
6.5.2.1	Unidad curricular: práctica Profesionalizante II	Pág.37
6.5.3	TERCER AÑO	Pág.38
6.5.3.1	Unidad curricular: práctica Profesionalizante III	Pág.38
CAPÍTULO VII. BIBLIOGRAFÍA GENERAL		Pág.40
CAPÍTULO VIII. GLOSARIO TÉCNICO		Pág.42

CAPÍTULO I

1. MARCO DE LA POLÍTICA EDUCATIVA PROVINCIAL Y NACIONAL PARA LA EDUCACIÓN TÉCNICO PROFESIONAL

1.1 La Educación Técnico Profesional en la provincia de Río Negro. Antecedentes y nuevos contextos

La Ley de Educación Nacional N° 26.206 define el Sistema Educativo Nacional en cuatro niveles –Educación Inicial, Educación Primaria, Educación Secundaria y Educación Superior– y ocho modalidades, entre las cuales se ubica como tal la Educación Técnico Profesional. Esta última se rige por las disposiciones de la Ley de Educación Técnico Profesional N° 26.058 y es la Educación Secundaria y la Educación Superior responsable de la formación de Técnicos Secundarios y Técnicos Superiores en áreas ocupacionales específicas y de la Formación Profesional.

En el ámbito provincial la sanción de la Ley Orgánica de Educación F N° 4819, establece en el Título 4, Cap. III, Art. 45, inc. b) “La habilitación de mecanismos administrativos e institucionales que permitan la elaboración de propuestas de formación de técnicos (...) superiores no universitarios como también de formación profesional en las diferentes áreas de la producción y los servicios, de acuerdo a las necesidades y potencialidades de desarrollo de cada región de la provincia”.

Toda Educación Técnico Profesional en el Nivel Superior tiene la intencionalidad de diseñar itinerarios profesionalizantes que permitan iniciar o continuar procesos formativos sistemáticos en diferentes áreas ocupacionales.

El ámbito y el nivel de Educación Superior se caracterizan por los rasgos peculiares de sus instituciones. Ellas deben no sólo desarrollar funciones relativas a la enseñanza sino también las concernientes a investigación, desarrollo, extensión y vinculación tecnológica e innovación. Dada la pluralidad de instituciones de Educación Superior que actualmente brindan, como parte de su oferta formativa la modalidad de Educación Técnico Profesional, se marca como orientación la progresiva especificidad de tales instituciones, de modo de asegurar las condiciones institucionales necesarias para que la expectativa señalada en el apartado anterior pueda concretarse.

En las últimas décadas en la Provincia de Río Negro se ha producido un incremento de propuestas formativas de Educación Superior Técnico Profesional que se implementan desde los Institutos Superiores, con anclaje territorial en diferentes puntos de la geografía provincial dando respuestas a las necesidades de formación e inserción laboral. En consonancia con lo que establece la Ley de Educación F N° 4819, en su Art. 55 establece que: “Las instituciones que brindan Formación Profesional deben reflejar en su propuesta de formación una estrecha vinculación con el medio productivo local y regional en el cual se encuentran insertas para dar respuesta a las demandas de calificación en aquellos sectores con crecimiento sostenido”.

Estas instituciones han transitado diferentes contextos, en cuanto a sus inicios y desarrollo, destacándose las acciones destinadas a cubrir el territorio y satisfacer las demandas de formación específica para el sector socio – productivo, promoviendo una adecuada

diversificación, que atiende a las expectativas y requerimientos de la estructura productiva, aprovechando en forma integral los recursos humanos, incrementando las posibilidades de actualización y reconversión para los integrantes del sistema y sus egresados.

De este modo identificamos momentos históricos vinculados a las primeras acciones, contándose con establecimientos que surgieron por medio de políticas nacionales (dependientes de la Ex DINEA), transfiriéndose luego al ámbito provincial; otros que se iniciaron a través de iniciativas locales (acuerdos interinstitucionales); aquellos que surgieron a partir de la reconversión institucional (establecimientos que constituían el sistema de formación Docente) y finalmente los establecimientos de creación reciente, que en muchos casos son propuestas ensambladas en los Institutos Superiores de Formación Docente. Asimismo cabe destacar el desarrollo de la Educación Privada cuya oferta educativa representa otra alternativa de Educación Superior Técnico Profesional ligada a diferentes campos del conocimiento, orientadas a la enseñanza aplicada y su vinculación con el empleo, y enfocada en el aumento de la demanda de formación, por parte de una población estudiantil cada vez más heterogénea, con particularidades asociadas al desarrollo económico y social.

CAPÍTULO II

2. FINALIDADES DE LA FORMACIÓN TÉCNICA EN DESARROLLO DE SOFTWARE FULL STACK

2.1 La Tecnicatura Superior en Desarrollo de Software Full Stack, aproximaciones a su campo de estudio conocimiento e impacto regional

El presente diseño curricular aspira a desarrollar, como lo expresa la Ley de Educación Técnico Profesional N° 26.058, una formación integral del/la estudiante a través de procesos que promuevan la construcción de múltiples habilidades cognitivas, socioemocionales, y técnico-profesionales en el área del desarrollo de software full stack, a fin de cimentar su futuro rol profesional, asegurando la articulación entre el sistema educativo y el sistema productivo de la región y del país.

Así, se asume la formación como un lugar desde donde poder potenciar a los sujetos, mejorando en primer lugar la calidad de vida y ofreciendo una posibilidad concreta de equidad social, achicando las brechas de vulnerabilidad, frente a la alta competitividad y la complejidad que asume el mundo del trabajo. La formación ya no puede ser concebida como una adquisición de aprendizajes fija y puntual que sirve para siempre, sino que debe constituir un proceso permanente que se renueva y se actualiza a lo largo de toda la vida profesional del sujeto; de modo que la formación inicial requiere prolongarse en la formación continua, a fin de que estos/as técnicos/as puedan ir adaptando sus cualificaciones profesionales a las nuevas especificaciones que se derivan de la evolución y el cambio tecnológico. Desde esta perspectiva se definen las siguientes finalidades formativas que tienden a formar técnicos/as superiores con capacidad para:

- Reconocer las prácticas y los procedimientos comunes en los entornos organizacionales para favorecer el desarrollo y administración del desarrollo de software.
- Operar de modo amplio y autónomo en el ámbito local y regional a través de la formación en el área de desarrollo de software.
- Gestionar saberes que permitan adaptarse a los rápidos adelantos de las tecnologías de la información y la comunicación y actuar con flexibilidad y disposición para aprender a desprender y aprender durante toda la vida.
- Desarrollar habilidades comunicativas, sociales y laborales que favorezcan el trabajo en equipo, promuevan la motivación y liderazgo a nivel personal y también se orienten al logro de los objetivos de la organización.
- Desarrollar capacidades emprendedoras freelancer requeridas para ser protagonistas de procesos de cambio, dirigidos a mejorar la empleabilidad, la productividad y la construcción de sus propios proyectos ocupacionales.
- Diseñar, producir o mantener (programar, adaptar e integrar) componentes o subconjuntos de software (clases, módulos, pantallas, rutinas, subsistemas, programas en general) conforme a especificaciones (funcionales y técnicas) para ser integrados en aplicaciones.
- Crear, mantener e implementar el código fuente que integra las aplicaciones para smartphones y otros dispositivos inteligentes. Desarrollar aplicaciones web.
- Planear, diseñar, implementar y mantener sitios web y aplicaciones móviles, codificando su lógica y configuración.

- Comprender y desarrollar tanto en la interfaz visible para los usuarios como en la lógica que opera detrás de un sitio o aplicación de Internet.
- Desenvolverse tanto en el frontend (parte visual) como en el backend (parte lógica) de un desarrollo web/móvil, poseer conocimientos y herramientas que le permiten afrontar cualquiera de las etapas de un proyecto web o móvil.
- Establecer estrategias tanto de cara al cliente como en el ámbito del servidor.
- Tener conocimientos sobre servidores, redes, data modeling, APIs, UI y UX y un buen entendimiento de las necesidades del cliente.
- Aplicar desde los fundamentos de la programación hasta el lanzamiento de aplicaciones web y móviles completas. Resolver problemas, desarrollando la lógica y usando las mejores prácticas de la industria.

En la era de la Economía del Conocimiento y la Industria 4.0 se requieren técnicos/as con habilidades específicas en programación, desarrollo de software y conocimiento sobre servidores, redes, base de datos, APIs, diseño UI y UX que la Tecnicatura Superior en Desarrollo de Software Full Stack ofrecerá. Esta es una propuesta superadora de formación superior acorde a las nuevas tecnologías, al interés común y a las necesidades regionales y nacionales.

El presente diseño curricular para la formación de Técnicos/as Superiores en Desarrollo de Software Full Stack se fundamenta en los Marcos de Referencia aprobados por Resolución del CFE N° 129/11, 351/19 y ha sido pensado y elaborado desde el diálogo y el consenso entre los/las docentes del instituto.

Un plan de estudio moderno, flexible y actualizado pero sin dejar de lado lo fundamental - aprender a programar- esto les permitirá a los/as estudiantes alcanzar una sólida formación en los conceptos de base que necesita un/a programador/a Full Stack, basado en Software Libre o de Código Abierto, con énfasis en el trabajo en equipo y aprendizajes basados en proyectos, formación continua, creatividad e innovación, capacidad de comunicación y una actitud emprendedora freelancer.

Este perfil híbrido entre el desarrollador frontend y backend es un experto con conocimientos en desarrollo web, desarrollo móvil, lenguajes de programación, base de datos, diseño UX/UI, API's, sistemas de control de versiones, redes y servidores web.

Esta propuesta aspira a fortalecer sus capacidades para ocupar empleos de calificación media o alta en el sector de la Industria 4.0 que cuenta con una amplia oferta laboral y buenas perspectivas de crecimiento, tanto a nivel global como en particular en las áreas: comercial, financiera, agro industrial, gas y petróleo; permitirá a los/las egresados/as insertarse en un mercado laboral amplio y diverso, con alta participación tanto de empresas grandes como pequeñas y medianas (PYMES), que requieren de la digitalización de sus procesos productivos y administrativos, a fin de mantener la competitividad en un mercado que de por sí es muy dinámico.

La propuesta curricular está organizada en tres años, compuestos por seis cuatrimestres de 16 semanas, con un total de 2048 horas reloj. La organización de la carrera contempla una progresiva complejidad de los campos del conocimiento general, de fundamentos, específicos y de la práctica profesionalizante.

2.3 Identificación del título

- Sector de la actividad socio-productiva: Informática/Nuevas Tecnologías
- Denominación del perfil profesional: Desarrollador de Software Full Stack
- Familia Profesional: Informática/Nuevas Tecnologías

2.4 Denominación del título

Técnico Superior en Desarrollo de Software Full Stack

2.5 Duración de la carrera en años

Tres años

2.6 Carga horaria de la carrera

2.048 hs

2.7 Objetivos de la carrera

La carrera tiene como objetivo garantizar la formación de profesionales que, independientemente de su lugar de residencia y de su contexto económico geográfico, puedan desempeñarse en un modelo productivo más innovador, competitivo e internacionalizado. De esta manera posicionará al/la técnico/a egresado/a como protagonista de un cambio de paradigmas que dan nacimiento a roles y habilidades digitales que presentan nuevas oportunidades de empleabilidad y desarrollo profesional:

- Ofrecer formación técnico profesional para el desarrollo de Software y su administración para dar respuesta a las necesidades del usuario local, regional y nacional.
- Conocer todas las facetas del frontend y del backend, los diferentes sistemas operativos y componentes que quedan en medio de esas fronteras.
- Proyectar, construir y comunicar un plan de producción de artefactos de software que satisfaga las demandas y necesidades especiales, funcionales y formales del cliente.
- Adquirir habilidades en el uso de herramientas entre las que se encuentran la programación, metodologías de desarrollo, servidores web, bases de datos, redes, data modeling, APIs y diseño UI/UX.
- Componer equipos de trabajo para la planificación de proyectos con profesionales afines a su tarea desde una visión totalizadora para el desarrollo de un software, la optimización del sistema, resguardo y seguridad de datos.
- Reconocer y cumplir las instrucciones establecidas en los protocolos de cada área de trabajo, descriptos como normas y procedimientos de trabajo.

2.8 Campo ocupacional

Puede desempeñarse en forma autónoma como freelancer, asumiendo casi la totalidad de las tareas propias del proceso backend / frontend trabajando en forma independiente resolviendo problemas de pequeñas organizaciones que requieren sistemas de baja complejidad y reducida dimensión.

Por otra parte, los Técnicos Superiores en Desarrollo de Software Full Stack o profesionales equivalentes con capacidad emprendedora pueden y suelen asociarse entre ellos/ellas para generar sus propias empresas para brindar servicios de desarrollo y proveer software a terceros; éstas son las nuevas empresas conocidas como unicornios o startup.

Ámbito Laboral:

El/la técnico/a podrá desempeñarse en organizaciones e instituciones:

- pequeñas, medianas o de mayor envergadura especializadas en el diseño y desarrollo de programas y aplicaciones informáticas.
- que provean software junto con otros servicios de asesoramiento y consultoría.
- dedicadas al desarrollo o mantenimiento de software propio o de terceros.
- como equipo de testers de sistemas de Software.
- equipos de IT de diferentes organismos públicos y privados.

Ámbito Profesional:

- Diseñar, programar, implementar y mantener aplicaciones Web y móvil frontend y backend.
- Realizar programación asincrónica y servicios web: AJAX, SOAP, Rest API.
- Desarrollar interfaces web responsivas para distintos tipos de pantallas.
- Programar con capacidad de conexión con un servidor de aplicaciones.
- Desarrollar Software y asesorar en servicios relacionados con la industria del software.
- Comercializar Software y realizar servicios de testeo de los mismos.
- Documentar y construir software según la problemática que aborda la aplicación.
- Diseñar, administrar e implementar bases de datos.
- Detectar, analizar y resolver problemas vinculados al Software que se insertará en un sistema.
- Diseñar e implementar Seguridad Web y Móvil.

2.9 Perfil del Egresado

El/a Desarrollador de Software Full Stack será un/a profesional programador con un perfil técnico muy completo que conoce cómo se diseña una aplicación web y una aplicación móvil y sabe programarlas, además de manejar distintos lenguajes de programación, sistemas operativos, bases de datos, diseño de interfaces, diseño UX/UI, servidores web, redes, sistema de control de versiones y APIs.

Los/las desarrolladores/as de este nivel tienen la particularidad de trabajar con la lógica compleja y lidiar con conceptos abstractos, conocen tanto lo referente al backend

(administración interna de una aplicación) y al frontend (parte visual). Cuentan con habilidades de comunicación efectiva en relación al resto del equipo.

El/la Técnico/a Superior en Desarrollo de Software Full Stack se encarga de manejar cada uno de los aspectos relacionados con la creación y el mantenimiento de una aplicación web y/o una aplicación móvil; FULL: lleno, STACK: pila, es un/a desarrollador/a que trabaja con pilas completas, desde el back hasta el front (una pila es un tipo de estructura de datos abstracta donde lo último que entra es lo primero que sale y el resto se apila).

2.10 Condiciones de ingreso

Teniendo en cuenta el Reglamento Académico Marco de la Jurisdicción, Resolución N.º 4077/14, en el mismo se establecen las siguientes condiciones:

- Artículo 5º: Ingreso.- A las instituciones de Educación Superior dependientes de la Provincia de Río Negro se ingresa de manera directa, atendiendo a la igualdad de oportunidades y la no discriminación.
- Artículo 6º: Inscripción.- Para inscribirse en una institución de Educación Superior es necesario presentar la siguiente documentación:

a) Solicitud de inscripción.

b) Constancia de estudios secundarios completos, acreditados con la presentación del título en original y copia, o constancia de título en trámite o constancia de finalización de cursado del secundario con materias adeudadas.

c) Fotocopia autenticada del documento de identidad (datos de identificación y domicilio).

d) Partida de Nacimiento actualizada, original y copia autenticada.

e) CUIL.

CAPÍTULO III

3. FUNDAMENTACIÓN PEDAGÓGICA DE LA PROPUESTA CURRICULAR

3.1 Acerca del Currículum, el Conocimiento, la Enseñanza y el Aprendizaje

La Educación Técnico- Profesional se constituye en un ámbito relevante de crecimiento y profesionalización en tiempos de avance científico-tecnológico. La formación de los profesionales se presenta como una necesidad. Esta modalidad establece una estrecha vinculación entre el campo educativo y el campo laboral y requiere de propuestas curriculares abiertas - flexibles en permanente actualización vinculadas con los contextos de actuación.

La Educación Técnico Profesional pretende una sólida formación teórico-práctica que incorpore el desarrollo de capacidades, destrezas en el manejo de la especificidad técnica para la que se forma y el aprendizaje de capacidades más integradoras que vinculen conocimientos con habilidades que apunten a la puesta en escena de herramientas de análisis en contextos, para la toma de decisiones. Esta visión intenta articular la formación de un profesional que aprenda no sólo aspectos instrumentales y técnicos sino contextuales frente a las demandas ocupacionales de la región. En este sentido la Ley de Educación Superior Nacional 24.521 en su Título II Capítulo 1 Artículo 3 explicita que :“ la educación superior tiene por finalidad proporcionar formación científica, profesional, humanística y técnica en el más alto nivel, contribuir a la preservación de la cultura nacional, promover la generación y desarrollo del conocimiento en todas sus formas, y desarrollar las actitudes y valores que requiere la formación de personas responsables, con conciencia ética y solidaria, reflexivas, críticas, capaces de mejorar la calidad de vida, consolidar el respeto al medio ambiente, a las instituciones de la República y a la vigencia del orden democrático”.

En este sentido se estructuran los Diseños Curriculares para la Formación Técnico Profesional y para comprenderlo es necesario explicitar las concepciones de currículum, conocimiento, enseñanza y aprendizaje que orientan el modo que asume el proceso formativo.

Todo currículum explicita ideas, conocimientos, posiciones acerca del hombre, la cultura y la sociedad. Se presenta como representación y concreción de un proyecto político-social-cultural en el que se manifiestan las intencionalidades educativas. El currículum se considera una herramienta de profesionalización en el sentido de promover capacidades en los ciudadanos para lograr progresos personales, profesionales y sociales. El currículum se concibe aquí como: “la expresión de un proyecto político, pedagógico y cultural, escenario de conflictos entre múltiples intereses e intencionalidades, es un eslabón entre la teoría educativa y la práctica pedagógica, entre lo que se afirma que puede ser la educación y lo que finalmente es. Por eso debe ser concebido como un proceso que involucra la práctica pedagógica misma como uno de sus componentes básicos.”¹

¹ Diseño Curricular de la Formación Docente (1988)

En el Currículum se seleccionan y sintetizan conocimientos que una sociedad define como válidos a ser transmitidos. Construidos y producidos socialmente que deben ser comprendidos en su carácter histórico y provisional. El conocimiento se presenta como un producto objetivado y contradictorio de procesos históricos sociales, culturales que son interpelados permanentemente por relaciones de poder.

En relación a los conocimientos a transmitir concebimos a la enseñanza como una práctica compleja que se sitúa en contextos configurados por múltiples dimensiones: institucionales, pedagógicas, didácticas, éticas, políticas. Su particularidad está definida por su relación específica con los conocimientos y los contextos de prácticas donde se producen.

La enseñanza implica siempre un intento deliberado y con relativa sistematicidad de transmisión de conocimientos. Toda enseñanza al ser una acción deliberada no es neutral sino que siempre se evidencia como una actividad política. Como toda acción intencional involucra aspectos personales, sociales y contextuales. En este sentido los aportes de las teorías psicológicas y sociológicas que toman lo grupal y lo social como intervinientes en el aprendizaje son un aporte relevante.

La enseñanza es un campo de prácticas que articula distintos ámbitos de decisión; el político, ético, epistemológico, el técnico y el de los contextos específicos de enseñanza. El desarrollo de una enseñanza situada requiere incorporar la dialéctica permanente entre los conocimientos y su transferibilidad en el ámbito de acción, esto exige la incorporación de diversos abordajes metodológicos según los campos de conocimientos que se involucren.

La intervención en la enseñanza involucra la relación con un aprendiz con sus características y las particularidades del conocimiento a transmitir. El aprendizaje hace referencia a los procesos a través de los cuales se adquieren los conocimientos. La enseñanza influye en el aprendizaje desde la tarea de aprendizaje que realiza el propio estudiante. Las situaciones de enseñanza que se proponen promueven procesos de aprendizaje y construcción de significados por parte del estudiante. Planteamos una concepción de aprendizaje como proceso de construcción de conocimientos que promueven la transferencia y comprensión de los mismos en situaciones cotidianas. Ubica al estudiante como un sujeto activo intelectualmente con posibilidades de aprendizaje personal y formado a través de la mediación con otros.

3.2 Consideraciones Metodológicas

El componente Metodológico requiere de la articulación entre el conocimiento como producción objetiva (lo epistemológico objetivo) y el conocimiento como problema de aprendizaje (lo epistemológico subjetivo) de este modo es concebido como un proceso de construcción. Lo metodológico implica tener en cuenta no solo la especificidad del contenido sino además una posición interrogativa frente a él necesaria para superar una postura instrumentalista y mecanicista en relación al método.

La construcción metodológica “se conforma a partir de la estructura conceptual (sintáctica y semántica) de la disciplina y la estructura cognitiva de los sujetos en situación de apropiarse de ella. Construcción de carácter singular que se genera en relación con un objeto de estudio

particular y con sujetos particulares. Se construye casuísticamente en relación con el contexto (áulico, institucional, social y cultural)”²

Una propuesta de enseñanza es un acto particular y creativo “de articulación entre la lógica disciplinar, las posibilidades de apropiación de ésta por parte de los sujetos y las situaciones y los contextos particulares que constituyen los ámbitos donde ambas lógicas se entrecruzan.”³

La propuesta metodológica deberá enfocarse a propiciar la confianza y seguridad en los estudiantes promoviendo actitudes de escucha respeto y comprensión que favorezcan la constitución de su identidad como futuros profesionales.

3.3 Acerca de la evaluación

Desde una racionalidad práctica y crítica, la evaluación constituye un campo conflictivo que nos involucra intersubjetivamente en tanto excede las cuestiones técnicas-pedagógicas por estar vinculada a cuestiones éticas, políticas, sociales e ideológicas. De allí la importancia de analizar y comprender los usos y sentidos de la evaluación, la finalidad que persigue, los intereses a los que responde y principios a los que adscribe.

Fernández Sierra⁴ plantea que la evaluación condiciona y determina la vida académica dentro y fuera del aula, involucrando todos los estamentos educativos, desde el trabajo de estudiantes y profesores hasta las decisiones políticas de más alto nivel.

Todo proceso de evaluación responde a múltiples dimensiones: las características de la institución, los proyectos institucionales, los estilos de gestión, las propuestas editoriales y curriculares, las particularidades de los docentes y de los estudiantes, entre otros.

En tal sentido, concebimos la evaluación como una práctica democrática y participativa abierta a la interrogación, la problematización, la búsqueda de entendimiento, la producción de conocimiento y la emancipación de todos los sujetos involucrados.

“La evaluación aplicada a la enseñanza y el aprendizaje consiste en un proceso sistemático y riguroso de recogida de datos, incorporado al proceso educativo desde su comienzo, de manera que sea posible disponer de información continua y significativa para conocer la situación, formar juicios de valor con respecto a ella y tomar las decisiones adecuadas para proseguir la actividad educativa mejorándola progresivamente”⁵

Esto nos lleva a afirmar la necesidad de revisar la definición de evaluación que suele sostenerse en los ámbitos educativos y que sólo la ligan a la constatación de conocimientos

² Edelstein, G. (1996): “Un capítulo pendiente: el método en el debate didáctico contemporáneo”. En: Corrientes Didácticas Contemporáneas. Buenos Aires: Paidós

³ Edelstein, G. Op Cit.

⁴ Fernández Sierra, J. (1994). “Evaluación del Currículum: perspectivas curriculares y enfoques en su evaluación”. En: Teoría del desarrollo del currículum. Málaga: Aljibe.

⁵ Casanova, M. A. (1995). Manual de evaluación educativa. La Muralla, Madrid.

aprendidos. La evaluación debe concebirse “desde su inclusión permanente y constante en nuestra cotidianeidad áulica y como una responsabilidad compartida”⁶

La evaluación como práctica de aprendizaje y de enseñanza promueve instancias de auto, co y heteroevaluación y no actividades instrumentales que generan medición y clasificación de los aprendizajes en los sujetos.

La evaluación como parte del proceso didáctico implica para los estudiantes una toma de conciencia de los aprendizajes adquiridos y, para los docentes, una interpretación de las implicancias de la enseñanza en esos aprendizajes. De este modo retroalimenta el proceso de enseñanza e informa a los estudiantes los progresos en sus aprendizajes. Será siempre formativa, motivadora, orientadora y al servicio de los protagonistas.

⁶ Ministerio de Educación (2009). Diseño Curricular para la Formación Docente de Nivel Primario. Subsecretaría de Formación y Capacitación Docente – Dirección de Nivel Superior. Río Negro.

CAPÍTULO IV

4. ORGANIZACIÓN CURRICULAR

4.1 Definición y caracterización de los campos de formación y sus relaciones

El Plan de Estudios se organiza en torno a cuatro campos de formación establecidos por la Resolución CFE N°229/14.

Campo de la formación general:

Destinado a abordar los saberes que posibiliten la participación activa, reflexiva y crítica en los diversos ámbitos de la vida laboral y sociocultural y el desarrollo de una actitud ética respecto del continuo cambio tecnológico y social.

Campo de la formación del fundamento:

Destinado a abordar los saberes científico tecnológicos y socioculturales que otorgan sostén a los conocimientos, habilidades, destrezas, valores y actitudes propios del campo profesional en cuestión.

Campo de la formación específica:

Dedicado a abordar los saberes propios de cada campo profesional, así como también la contextualización de los desarrollados en la formación de fundamento.

Campo de la formación en la práctica profesional:

Destinado a posibilitar la integración y contrastación de los saberes construidos en la formación de los campos descriptos, y garantizar la articulación teoría-práctica en los procesos formativos a través del acercamiento de los estudiantes a situaciones reales de trabajo.

4.2 Carga horaria por campo (*)

Campos de Formación	Porcentaje en Plan de estudios	Porcentaje Actividades teóricas	Porcentaje Actividades prácticas formativas	Total de horas de la carrera 2048 horas reloj
Formación general (Porcentaje mínimo: 5%)	12,5%	70%	30%	
Formación de fundamento (Porcentaje mínimo: 20%)	28,125%	50%	50%	
Formación específica (Porcentaje mínimo: 30 %)	46,875%	40%	60%	
Prácticas Profesionalizantes (Porcentaje mínimo: 20%)	12,5%	20%	80%	

(*) Según lo establecido por la Resolución N° 229/14 del Consejo Federal de Educación.

4.3 Definición de los formatos curriculares que integran la propuesta

Unidades curriculares: Se entiende por “unidad curricular” a aquellas instancias curriculares que, adoptando distintas modalidades o formatos pedagógicos, forman parte constitutiva del plan, organizan la enseñanza y los distintos contenidos de la formación y deben ser acreditadas por los estudiantes. Las mismas se distribuyen de la siguiente manera:

- **Las asignaturas:** son unidades curriculares definidas por la enseñanza de marcos disciplinares o multidisciplinarios y sus derivaciones metodológicas para la intervención educativa. Son de valor troncal para la formación y se caracterizan por brindar conocimientos, modos de pensamiento y modelos explicativos. Permiten el análisis de problemas, la investigación documental, la preparación de informes, el desarrollo de la comunicación oral y escrita y la aproximación a métodos de trabajo intelectual transferibles a la acción profesional. En cuanto al tiempo y ritmo, pueden adoptar la periodización anual o cuatrimestral, incluyendo su secuencia en cuatrimestres sucesivos.
- **El trabajo de campo:** propone un acercamiento real al contexto, a la cultura de la comunidad, a las instituciones y los sujetos en los que acontecen las experiencias de práctica. Los Trabajos de Campo constituyen espacios sistemáticos de trabajos de indagación en terreno e intervenciones en espacios acotados, desarrollo de micro experiencias, prácticas sistemáticas en contextos específicos y diversos.
- **Los Talleres** son unidades curriculares que promueven la resolución práctica de situaciones que requieren de un hacer creativo y reflexivo, poniendo en juego marcos conceptuales disponibles, también posibilita la búsqueda de otros marcos necesarios para orientar, resolver o interpretar los desafíos de la producción. Como modalidad pedagógica, el taller apunta al desarrollo de capacidades para el análisis de casos y de alternativas de acción, la toma de decisiones y la producción de soluciones e innovaciones. En este proceso, se estimula la capacidad de intercambio, la búsqueda de soluciones originales y la autonomía del grupo. Su organización es adaptable a los tiempos cuatrimestrales.
- **Los seminarios** son unidades que se organizan en torno a casos, problemas, temas o corrientes de pensamientos para cuyo análisis se requiere de una producción específica, la contraposición de enfoques, posiciones y debate. Implican instancias académicas de estudio de problemas relevantes para la formación profesional, a través de la reflexión crítica de las concepciones o supuestos previos sobre tales problemas, que los estudiantes tienen incorporados como resultado de su propia experiencia, para luego profundizar su comprensión a través de la lectura y el debate de materiales bibliográficos o de investigación. Estas unidades permiten el cuestionamiento del "pensamiento práctico" y ejercitan en el trabajo reflexivo y el manejo de literatura específica, como usuarios activos de la producción del conocimiento. Los seminarios se adaptan bien a la organización cuatrimestral, atendiendo a la necesidad de organizarlos por temas/ problemas. Asimismo estos espacios incluyen dispositivos que enriquecen el proceso formativo, como propuestas de opcionalidad académica que supondrán la articulación entre diferentes instituciones (sociales, académicas, políticas, etc.).
- **Conferencias y coloquios** conforman encuentros de aprendizaje con especialistas especialmente invitados, sobre temáticas relativas a los contenidos que se están desarrollando en los distintos cursos para resignificar, ampliar y profundizar los marcos interpretativos.
- **Seminarios de intercambio y debate de experiencias:** encuentros de presentación de experiencias, de informes de estudios de campo, de trabajos monográficos, posters, y otras modalidades, con debate de sus desarrollos y conclusiones con el propósito de valorizar, producir, sistematizar y socializar conocimientos, investigaciones operativas llevadas a cabo por los estudiantes durante su proceso de formación.

- **Congresos, Jornadas, Talleres:** actividades académicas sistematizadas que organizadas por los Institutos Superiores u otro tipo de instituciones reconocidas permiten, aún antes del egreso, vincular a los estudiantes con el mundo técnico-profesional.

CAPÍTULO V.

5. ESTRUCTURA CURRICULAR

5.1 Mapa curricular

TECNICATURA SUPERIOR EN DESARROLLO DE SOFTWARE FULL STACK							
Formación General		Formación de Fundamento		Formación Específica		Prácticas Profesionalizantes	
PRIMER AÑO							
Primer Cuatrimestre	Segundo Cuatrimestre	Primer Cuatrimestre	Segundo Cuatrimestre	Primer Cuatrimestre	Segundo Cuatrimestre	Primer Cuatrimestre	Segundo Cuatrimestre
Inglés Técnico I (4hs/64hs)		Matemática (4hs/128hs)		Programación (4hs/128hs)			Práct. Prof. (4hs/64hs)
		Laboratorio Full Stack I (4hs/128hs)		Base de Datos (4hs/64hs)	Diseño UX/UI (4hs/64hs)		
		Arquitectura de las Computadoras (4hs/64hs)					
SEGUNDO AÑO							
Primer Cuatrimestre	Segundo Cuatrimestre	Primer Cuatrimestre	Segundo Cuatrimestre	Primer Cuatrimestre	Segundo Cuatrimestre	Primer Cuatrimestre	Segundo Cuatrimestre
Inglés Técnico II (4hs/64hs)		Laboratorio Full Stack II (4hs/128hs)		Programación Backend (4hs/128hs)			Práct. Prof. (4hs/64hs)
		Integración de Aplicaciones (4hs/64hs)		Desarrollo de Software (4hs/128hs) Programación Frontend (4hs/64hs)			
TERCER AÑO							
Primer Cuatrimestre	Segundo Cuatrimestre	Primer Cuatrimestre	Segundo Cuatrimestre	Primer Cuatrimestre	Segundo Cuatrimestre	Primer Cuatrimestre	Segundo Cuatrimestre
InglésTécnico III (4hs/64hs)	Ética y Responsabilidad Social (4hs/64hs)	Innovación y Desarrollo Emprendedor (4hs/64hs)		Desarrollo Web (4hs/128hs)			Práctica Profesionalizante (4hs/128hs)
				Desarrollo Móvil (4hs/128hs) Gestión de Proyectos de Software (4hs/128hs)			
Total horas Formación General: 256 hs		Total horas Formación Fundamento: 576 hs		Total horas Formación Específica: 960 hs		Total horas Prácticas Prof: 256 hs	
TOTAL HORAS DE LA CARRERA: 2048 hs.							

CAPÍTULO VI

6. UNIDADES CURRICULARES

6.1 UNIDADES CURRICULARES.

Componentes Básicos En la presentación de las unidades curriculares se explicitan los siguientes componentes: formato, orientaciones para la selección de contenidos, y bibliografía.

- **Formatos.** Implican no solo un determinado modo de transmisión del conocimiento, sino también una forma particular de intervención en los modos de pensamiento, en las formas de indagación, en los hábitos que se construyen para definir la vinculación con un objeto de conocimiento. (Resolución CFE N° 24/07).
- **Finalidades formativas de una unidad curricular.** Las finalidades formativas de la unidad curricular son un componente del Encuadre Didáctico de los Diseños Curriculares Provinciales. Son un tipo particular de propósitos que refieren a aquellos saberes que los estudiantes deben acreditar al finalizar el curso y que están vinculados a las prácticas profesionalizantes.
- **Ejes de contenidos descriptores.** Adscribiendo a la concepción de los diseños curriculares como un “marco de organización y de actuación y no un esquema rígido de desarrollo”, el presente diseño curricular incorpora criterios de apertura y flexibilidad para que “el currículum en acción” adquiera una fluida dinámica, sin que sea una rígida e irreflexiva aplicación del diseño curricular o un requerimiento burocrático a ser evitado.”. En ese encuadre, se presentan los ejes de contenidos, concebidos como las nociones más generales y abarcadoras que constituirán la unidad curricular con la función de delimitar, definir y especificar los campos conceptuales que la integran.

6.2 CAMPO DE LA FORMACIÓN GENERAL

6.2.1 PRIMER AÑO

▪ 6.2.1.1 Inglés Técnico I

Formato: Asignatura

Régimen de cursada: Cuatrimestral

Ubicación en el diseño curricular: 1er año 1er cuatrimestre

Asignación de horas semanales: 4 hs.

Total de horas: 64 hs.

Finalidades formativas de la Unidad Curricular

Adquirir nociones básicas del lenguaje, tanto oral como escrito, posibilitando el acceso a bibliografía en inglés específica para el desarrollo de software, logrando autonomía en la lectura e interpretación de textos técnicos.

Ejes de contenidos. Descriptores

Lectura e interpretación de textos. Vocabulario. Lectura de información técnica. Comprender textos de complejidad creciente. Redactar textos para solicitar información técnica por email o en foros. Elaboración de listas de materiales. Reconocimiento de abreviaturas.

6.2.2 SEGUNDO AÑO

▪ **6.2.2.1 Inglés Técnico II**

Formato: Asignatura

Régimen de cursada: Cuatrimestral

Ubicación en el diseño curricular: 2do año 1er cuatrimestre

Asignación de horas semanales: 4 hs.

Total de horas: 64 hs.

Finalidades formativas de la Unidad Curricular

Afianzar los conocimientos sobre la lengua tanto escrita como oral para lograr manejo de los textos empleados en desarrollo de software y los sistemas de signos que permiten la comunicación entre individuos de distintas nacionalidades.

Ejes de contenidos. Descriptores

Lectura y escritura de textos en inglés. Vocabulario específico. Sustantivos Compuestos. Características del texto escrito. Elementos de Cohesión. Coherencia semántica, contraste y énfasis en el modo escrito. Vocabulario de inglés técnico y abreviaturas propias del lenguaje del Software.

6.2.3 TERCER AÑO

▪ **6.2.3.1 Ética y Responsabilidad Social**

Formato: Asignatura

Régimen de cursada: Cuatrimestral

Ubicación en el diseño curricular: 3er año 2do cuatrimestre

Asignación de horas semanales: 4 hs.

Total de horas: 64 hs.

Finalidades formativas de la Unidad Curricular

Promover la reflexión respecto de las cuestiones de índole valorativa que surgen en los diversos desempeños profesionales; la responsabilidad que requiere la creación y comercialización de Software, el respeto por su equipo de trabajo y por el medio ambiente en el que convive y para el que desarrolla su tarea. Interpretar los derechos constitucionales y sus obligaciones como ciudadanos.

Ejes de contenidos. Descriptores

Definición de ética. Características. División. El hombre como ser ético y social. El hombre y la organización: relación entre el hombre y la organización en la teoría de la Administración. La dimensión ética del comportamiento. Ética y moral. Relativismo y absolutismo ético. Ética como filosofía moral. Teorías normativas. Ética aplicada. La ética empresarial. El marco ético de la empresa. Responsabilidades empresariales, sociales y del impacto al medio ambiente. Elaboración de un código de ética, reportes de sustentabilidad, valoración ética de la empresa.

La ética en el desarrollo de un Software. Valor de la información. Bases de datos públicas y privadas. Propiedad de datos empresarios. Secretos comerciales e industriales. Paradigmas de la informática. Paradigma para la construcción, producción y distribución del software. Software privativo, abierto y libre.

▪ **6.2.3.2 Inglés Técnico III**

Formato: asignatura.

Régimen de cursada: cuatrimestral.

Ubicación en el diseño curricular: 3er año 1er cuatrimestre

Cantidad de horas semanales: 4 hs.

Total de horas: 64 hs.

Finalidades formativas de la Unidad Curricular

Este espacio curricular brindará las herramientas fundamentales para que el técnico emplee la lengua extranjera con el propósito de consultar bibliografía específica. Comprender textos orales y

escritos utilizando las informaciones tanto generales como específicas que proporcionan dichos textos.

Ejes de contenidos. Descriptores

Referencias contextuales. Conectores básicos. Organización de la información. Funciones del lenguaje. Prefijos y sufijos. Comparaciones. Voz pasiva. Sustantivos compuestos. Ejemplificaciones. Definiciones. Reformulaciones. Causa-efecto. Predicciones. Referencia. Obligación. Posibilidad. Condiciones. Vocabulario específico de informática. Preposiciones. Descripciones. Sustantivos compuestos ejemplificaciones.

6.3 CAMPO DE LA FORMACIÓN DE FUNDAMENTO

6.3.1 PRIMER AÑO

▪ 6.3.1.1 Matemática

Formato: asignatura.

Régimen de cursada: anual

Ubicación en el diseño curricular: 1er año

Cantidad de horas semanales: 4 hs.

Total de horas: 128 hs.

Finalidades formativas de la Unidad Curricular

Contar con los conocimientos, habilidades y destrezas básicas y necesarias para que desarrolle su capacidad de razonamiento que le permitirán resolver situaciones problemáticas con base en el pensamiento simbólico para el desarrollo de Software.

Ejes de contenidos. Descriptores

Funciones, inyectivas, sobreyectivas y biyectivas. Relaciones; reflexividad, simetría, transitividad, equivalencia. Conjuntos; diagramas de Venn, operaciones, complementos, producto cartesiano, conjunto potencia. Aritmética modular. Relaciones de congruencia. Sistemas de numeración. Elementos de lógica. Lógica proposicional, conectivos lógicos. Tablas de verdad. Formas normales; conjuntiva y disyuntiva. Validez. Lógica de predicados; cuantificadores universal y existencial. Modus ponens y modus tollens. Limitaciones de la lógica de predicados. Lógica matemática, manejo de vectores y matrices. Sistema binario y hexadecimal.

▪ **6.3.1.2 Laboratorio Full Stack I**

Formato: asignatura.

Régimen de cursada: anual

Ubicación en el diseño curricular: 1er año

Cantidad de horas semanales: 4 hs.

Total de horas: 128 hs.

Finalidades formativas de la Unidad Curricular

Este espacio curricular brindará las herramientas fundamentales para la instalación y administración de servidores virtualizados locales y en nube, conceptos de hosting, gestión de repositorios y control de versiones. Les acercará las herramientas básicas que le permitan instalar framework, IDEs, CMS y CMRs. Desarrollará un servidor virtual que integrará todos los conceptos vistos.

Ejes de contenidos. Descriptores

Introducción al Software Libre y GNU-Linux. Terminales Shell. Scripting. Comandos básicos. Introducción a Redes y protocolos de Red. Direccionamiento IP. Modelo de OSI, TCP-IP. Interfaces. Introducción a la Virtualización. Instalación de Servidores web: LAMP y Nginx. Instalación de herramientas y aplicaciones nativas y web con Software Libre y Open Source: CMS, CMR, DNS, FTP, Administrador BD. Git y GitHub: Creación de repositorios y ramas, Implementación de Grids y Grids Mobile First, Subida al servidor - Configuración del hosting y acceso al servidor/host. Contenedores: Introducción a Docker, Arquitectura, Implementación.

▪ **6.3.1.3 Arquitectura de las Computadoras**

Formato: asignatura.

Régimen de cursada: cuatrimestras

Ubicación en el diseño curricular: 1er año 1er cuatrimestre

Cantidad de horas semanales: 4 hs.

Total de horas: 64 hs.

Finalidades formativas de la Unidad Curricular

Este espacio curricular brindará las herramientas fundamentales para conocer y desarrollar los elementos de un sistema de computación. Comprender los sistemas de representación de la información. Desarrollar la estructura, organización y funcionamiento de los procesadores, memoria y dispositivos de una computadora. Comprender los dispositivos y elementos de interconexión de un sistema informático.

Ejes de contenidos. Descriptores

Elementos de una computadora, arquitectura Von Neumann: clasificación de las computadoras, introducción a los circuitos lógicos. Representación de la información: alfanuméricos, numéricos, ASCII, punto fijo y flotante. Conceptos de sistemas operativos: GNU-Linux, Unix, Windows. Niveles de lenguajes de programación. Lenguaje Assembler. Lenguajes de alto nivel. Microprogramación. Sistemas con interfaces gráficas. Compiladores, interpretadores, ensambladores, vinculadores, depuradores, editores. Redes de computadoras. Bus de datos, memorias, tipos, circuito de carga y ejecución.

6.3.2 SEGUNDO AÑO

▪ 6.3.2.1 Laboratorio Full Stack II

Formato: asignatura.

Régimen de cursada: anual

Ubicación en el diseño curricular: 2do año

Cantidad de horas semanales: 4 hs.

Total de horas: 128 hs.

Finalidades formativas de la Unidad Curricular

Integrar y ampliar sus conocimientos teórico-prácticos con el armado de servidores, base de datos, FTP. Lograr habilidades para diseñar y construir aplicaciones sencillas para el acceso web en un ambiente cliente-servidor donde se debe considerar e implementar aspectos de seguridad relativos a las mismas.

Ejes de contenidos. Descriptores

Aplicaciones para redes. Protocolos a nivel de la capa de aplicación. Interfaces web: “browsers” y APIs. Subprotocolos TCP y UDP. El “socket” como abstracción. Modelo cliente servidor. Primeras acciones de ambos. Creación de “sockets” y ligado de direcciones. Par cliente/servidor TCP. Concepto de sesión. Par cliente/servidor UDP. Concepto de paquete. “Polling” con primitivas bloqueantes. RCP. Aspectos de administración de redes, uso de contraseñas y mecanismos de control de acceso, servidores de nombres de dominios y de servicios, proveedores de servicios en

Internet. Aspectos de seguridad y firewalls. Asuntos de calidad de servicio: comportamiento, recuperación de fallos. Tendencias técnicas que afectan la seguridad del software. Bugs. Autenticación y autorización.

▪ **6.3.2.2 Integración de Aplicaciones**

Formato: asignatura.

Régimen de cursada: cuatrimestral

Ubicación en el diseño curricular: 2do año 1er cuatrimestre

Cantidad de horas semanales: 4 hs.

Total de horas: 64 hs.

Finalidades formativas de la Unidad Curricular

Este espacio curricular brindará las herramientas fundamentales para la integración de aplicaciones, sus casos de uso, los sistemas de software distribuido, la arquitectura Rest, su desarrollo teniendo en cuenta los acuerdos y restricciones definidos por el protocolo HTTP, permitir que otras aplicaciones se comuniquen para crear, leer, actualizar o eliminar información, conectar, integrar y extender sistemas de software. Comunicaciones con los servidores web.

Ejes de contenidos. Descriptores

Programación asincrónica y servicios web: AJAX, SOAP, Rest. Definición de API, la interfaz, reusabilidad. Arquitectura Rest, Definición y Fundamentos, Clientes-Servidor. Recursos, dirección de un recurso (URI), Categorías de los Recursos: colección, instancia, controlador. Recursos Uniformes y URI. Creación y Actualización de un Recurso, Formularios. Representaciones y Parámetros: JSON y XML. Interfaz Uniforme de HTTP: Las Operaciones CRUD, métodos de HTTP POST(Create), GET(Reading), PUT(Updating) y DELETE(Deleting). Parámetros de la Cabecera de Solicitud y Parámetros de la Cabecera de Respuesta. Códigos de Estado. Seguridad en un API REST: Mecanismos, Llaves de API, HTTP Basic, HTTP Digest, OAuth.

6.3.3 TERCER AÑO

▪ **6.3.3.1 Innovación y Desarrollo Emprendedor**

Formato: asignatura.

Régimen de cursada: cuatrimestral

Ubicación en el diseño curricular: 2do año 1er cuatrimestre

Cantidad de horas semanales: 4 hs.

Total de horas: 64 hs.

Finalidades formativas de la Unidad Curricular

Comprender la innovación como un proceso y como una solución a las preocupaciones de las personas; el emprendedorismo como procedimiento para bosquejar ideas de negocios, formular proyectos para materializarlas y la implementación de los mismos de manera de impactar en la competitividad de las organizaciones de la región.

Ejes de contenidos. Descriptores

El desarrollo sostenible. Las personas emprendedoras, características, creativas e innovadores, las decisiones. El empresario emprendedor. Dinamizador social. La empresa y el entorno empresarial. Concepto de creatividad e innovación. Técnicas de creatividad e innovación aplicada al puesto de trabajo o al nuevo emprendimiento. La innovación como solución. Diseñando el prototipo de emprendimiento. Técnicas para ayudar a definir modelos de negocios. Diseño del modelo de negocios: empresa y proceso. Modelo de negocio. Ontología de modelos de negocio. Los bloques del modelo de negocio: describir, analizar y diseñar emprendimientos. Estrategia de un modelo de negocio. Organización jurídica de nuevos emprendimientos. La venta de las ideas de negocios. Plan de negocio, sus partes. Introducción a software para la redacción de un plan de negocio. Gestión de costos de software.

6.4 CAMPO DE LA FORMACIÓN ESPECÍFICA

6.4.1 PRIMER AÑO

▪ 6.4.1.1 Programación

Formato: asignatura

Régimen de cursada: anual

Ubicación en el diseño curricular: 1er año

Cantidad de horas semanales: 4 hs

Total de horas: 128 hs

Finalidades formativas de la Unidad Curricular

Desarrollar habilidades, capacidades y destrezas como instrumentos para el desarrollo de la capacidad de razonamiento y de resolución de problemas para fortalecer las bases necesarias para el pensamiento computacional. Diseñar algoritmos eficientes para la solución de problemas computacionales. Diseñar las estrategias de prueba para validación de algoritmos y de programas. Programación, Programación funcional, Programación Imperativa y Programación orientada a objetos.

Ejes de contenidos. Descriptores

Algoritmos. Concepto de algoritmo, resolución algorítmica de problemas, estrategias de diseño, de implementación, de depuración. Algoritmos fundamentales, algoritmos numéricos simples. Estructuras fundamentales, variables, tipos, expresiones y asignaciones, entrada/salida, estructuras de control condicionales e iterativas, funciones y pasaje de parámetros, descomposición estructurada. Introducción a lenguajes de programación. Representación de datos numéricos, rango, precisión y errores de redondeo. Arreglos. Representación de datos de caracteres, listas y su procesamiento. Recolección de espacios no utilizados. La elección de una estructura de datos adecuada. Diferentes paradigmas. Paradigma funcional, lógico, imperativo, características fundamentales. Estructuras de datos y control. Diseño orientado a objetos. Diseño estructurado. Diseño orientado al reuso de componentes, incorporación de elementos disponibles al diseño. Diseño de interfaces con el usuario. Prototipos rápidos. Diseño orientado a objetos, encapsulamiento y ocultamiento de información, separación entre comportamiento e implementación, clases y subclasses. Distinción entre validación y verificación. Enfoques estáticos y dinámicos. Fundamentos de testeo, de caja negra y blanca. Pruebas funcionales y estructurales. Otro tipo de objetivos: verificación de usabilidad, confiabilidad, seguridad. Registro de fallas, seguimiento de fallas e informes técnicos. Prueba unitaria, de integración, validación y prueba del sistema. Desarrollo conducido por el testeo. Refactorización del código. Testeo de regresión. Verificación y validación de artefactos que no constituyen código: documentación, archivos de ayuda, material de capacitación. Inspecciones, revisiones cruzadas, auditorías. Herramientas para desarrollo en ambientes web. frontend.

▪ 6.4.1.2 Base de Datos

Formato: Asignatura

Régimen de cursada: Cuatrimestral

Ubicación en el diseño curricular: 1er año 1er cuatrimestre

Asignación de horas semanales: 4 hs.

Total de horas: 64 hs.

Finalidades formativas de la Unidad Curricular

Comprender y utilizar sistemas de bases de datos que faciliten el almacenamiento y administración de la información, garantizando su integridad, accesibilidad y seguridad. Diseñar bases de datos relacionales. Definir la estructura, índices y relaciones entre tablas de bases de datos para la manipulación y actualización de los datos almacenados. Optimizar bases de datos, mediante procedimientos de normalización. Analizar y diseñar una base de datos adecuada a la situación planteada.

Ejes de contenidos. Descriptores

Concepto de base de datos, tipos de bases de datos. Entidad. Relaciones entre entidades. Atributo de las entidades. Clave principal y clave secundaria. Cardinalidad de las relaciones. Modelos de datos. Estructuras de almacenamiento. Modelo entidad-relación. Bases de datos no relacionales (ej. orientadas a documentos, orientadas a objetos, orientadas a grafos). Consultas y operaciones Diagrama de Entidad-Relación, MER-DER. El modelo relacional, transformación del modelo E/R a relacional. Concepto y evolución de los sistemas de bases de datos, sus componentes, funciones de un sistema de base de datos. Concepto de almacenamiento y recuperación de información, captura y representación, aplicaciones, búsqueda, recuperación, vinculación, navegación. Metadatos o esquemas asociados con los datos objeto del procesamiento. Problemas de escalabilidad, eficiencia y efectividad. Privacidad, integridad, seguridad y preservación de la información. La persistencia e integridad de los datos. Diseño de bases de datos, dependencia funcional, formas normales, descomposición de un esquema, claves primarias y secundarias. Procesamiento de transacciones, fallas y recuperación, control de concurrencia. Bases de datos distribuidas, problemas que surgen con su explotación.

▪ 6.4.1.3 Diseño UX - UI

Formato: Asignatura

Régimen de cursada: Cuatrimestral

Ubicación en el diseño curricular: 1er año 2do cuatrimestre

Asignación de horas semanales: 4 hs.

Total de horas: 64 hs.

Finalidades formativas de la Unidad Curricular

Reconocer la importancia del diseño UX-UI para optimizar la interacción de las personas con el producto, la usabilidad, accesibilidad y la interacción. Generar una experiencia que se vuelva fluida y casi intuitiva. Incorporar elementos visuales identificatorios propios de la organización o marca y a través de la interacción entre el usuario y dicha interfaz, contribuir al posicionamiento y generar experiencias positivas en relación a esta.

Ejes de contenidos. Descriptores

Incorporar los conceptos de los atributos de calidad, la Usabilidad, Aprendizaje, Eficiencia, Memorabilidad, Errores, Satisfacción. Conceptos y diferencias en UX (experiencia de usuario) y Usabilidad. Técnicas de medición y uso de una interfaz. Herramientas de medición y análisis de resultados para medir la experiencia de usuario. Concepto de arquitectura de información

mediante el diseño de patrones para un sitio. Características del diseño centrado en el usuario. La accesibilidad: Capacidad de interacción de la persona con la interfaz, considera tanto el contexto de uso, como las propias aptitudes del usuario. Interacción: entre la interfaz y el usuario. Instancias y calidad del prototipo. Interfaces. Prototipos digitales. Patrones de interacción y de navegación. Metodología design thinking: definir un problema, formular hipótesis e ideas, testearlas a través de prototipos o MVPs. Diseño UI: Sistemas de grillas, columnas, cuadrículas, layouts. Principios de Heurísticas. Definición y armado de Moodboards. Human Interface y Material Design. Contrastes y legibilidad. Microinteracciones y Motion.

6.4.2 SEGUNDO AÑO

▪ 6.4.2.1 Desarrollo de Software

Formato: Asignatura

Régimen de cursada: Anual

Ubicación en el diseño curricular: 2do año

Asignación de horas semanales: 4 hs.

Total de horas: 128 hs.

Finalidades formativas de la Unidad Curricular

Afianzar los conocimientos de los paradigmas de programación vistos en el primer año y llevarlo a la práctica mediante la codificación en un lenguaje de programación, desarrollando su capacidad de abstracción. Lograr autonomía para revisar y corregir programas dados, resolver diversos tipos de problemas, diseñar una solución, codificar e integrar partes de código utilizando el ambiente de programación.

Ejes de contenidos. Descriptores

Programación orientada a objetos: clases, objetos y métodos. El Polimorfismo, sobrecarga de operadores binarios. Patrones. Implementación de los conceptos en distintos lenguajes. Programación orientada a eventos. Utilización del Sistemas de control de versiones: git. Algoritmos de búsqueda sucesiva y binaria, de ordenamiento con tiempos cuadráticos. Tablas de hashing, Árboles de búsqueda binaria. Representación de grafos. Algoritmos de camino mínimo. Concepto de autómatas. Elementos de complejidad de algoritmos. Colas y Pilas. Programación defensiva: alternativas o dispositivos de lenguajes de programación para evitarlo. Proceso de desarrollo centrado en el usuario: foco temprano en los usuarios, prueba empírica de la calidad, diseño iterativo. Medidas de evaluación: utilidad, eficiencia, facilidad de aprendizaje, satisfacción del usuario. Modelos de diseño de la interacción: atención, movimiento, cognición, percepción y reconocimiento. Análisis orientado a objetos y UML. Herramientas de modelización UML. Diseño de Sistemas.

▪ **6.4.2.2 Programación Backend**

Formato: Asignatura

Régimen de cursada: Anual

Ubicación en el diseño curricular: 2do año

Asignación de horas semanales: 4 hs.

Total de horas: 128 hs.

Finalidades formativas de la Unidad Curricular

Acercar las herramientas para que el estudiante diseñe la lógica y las soluciones para que todas las acciones solicitadas por el frontend sean ejecutadas de manera correcta en el backend. Lleve a cabo proyectos con la utilización de métodos y la ayuda de herramientas propias de su ámbito de acción. Reconocer, diferenciar, documentar y validar las necesidades que justifican sus proyectos según los lenguajes de programación utilizados, incluyendo su base de datos.

Ejes de contenidos. Descriptores

Conceptos de la programación Backend. Lenguajes de Programación Backend. Relación con el diseño y la arquitectura de software. Paradigmas, métodos y herramientas: Modelos de programación backend. Desarrollo para móviles multi plataforma. Análisis de requerimientos de software, el proceso, partes interesadas. Requerimientos funcionales, prioridades y criterios de realización. Diagramas de clase. Escenarios, historias y casos de uso; diseño centrado en el usuario. Representación del comportamiento: diagramas de secuencia, máquinas de estado, diagramas de actividad. Uso de librerías del servidor. Base de Datos relacionales y no relacionales, almacenamiento en memoria Redis y Memcached. Almacenamiento web: sesiones, cookies, caché web browser. Conexión con bases de datos. Análisis de datos: datos de referencia y de operaciones; datos de nivel de recursos y de volumen de actividad. Diccionario de datos. Requerimientos no funcionales, ejemplos y su influencia en el diseño del software. Estándares de documentos de requerimientos.

▪ **6.4.2.3 Programación Frontend**

Formato: Asignatura

Régimen de cursada: Cuatrimestral

Ubicación en el diseño curricular: 2do año 2do cuatrimestre

Asignación de horas semanales: 4 hs.

Total de horas: 64 hs.

Finalidades formativas de la Unidad Curricular

Adquirir las habilidades necesarias para desarrollar la parte visual de una aplicación móvil o una web. Diseñar la estructura, la tipografía, la colorimetría, insertar imágenes, banners, documentos, formularios. Desarrollar un diseño atractivo e intuitivo para mejorar la experiencia del usuario. Uso de tecnologías de diseño y desarrollo web que corren en el navegador y que se encargan de la interactividad con los usuarios.

Ejes de contenidos. Descriptores

Conceptos básicos de aplicaciones web y móviles. Lenguajes de programación web: html, css, php y javascript. Maquetación de aplicaciones multiplataforma. CMS. Creación de hojas de estilo CSS. Adjuntar hoja de estilos. Diseño por capas. Modelo de caja CSS. Lenguaje de desarrollo web. Composición funcional, herencia prototípica, cierres, delegación de eventos, flujo de control asíncronico, utilización de Marcos Populares y Marcos de Prueba. Scripting. Código HTML. Estructura y etiquetas HTML. Formularios HTML. Imágenes. Maquetación web. Controles HTML. Controles Servidor. Uso de lenguaje script. Diseño. Método de acceso a datos. Hojas de estilo. Estructura y sintaxis de un archivo Script. Variables, arreglos, objetos. Funciones. Modelo de Objetos del Documento (DOM). Nodos: manipulación, recorrido. Eventos. Ventanas de diálogo. Depuración de errores. Validación de datos de formularios. Formato JSON para intercambio de datos e instrucciones para su manipulación. XML. Frameworks, librerías o preprocesadores para crear todo tipo de interfaces de usuario.

6.4.3 TERCER AÑO**▪ 6.4.3.1 Desarrollo Web**

Formato: Asignatura

Régimen de cursada: Anual

Ubicación en el diseño curricular: 3er año

Asignación de horas semanales: 4 hs.

Total de horas: 128 hs.

Finalidades formativas de la Unidad Curricular

Ampliar y aplicar los saberes adquiridos sobre la aplicación de algunos de los paradigmas de programación y llevarlo a la práctica mediante la codificación en un lenguaje de programación, utilizando IDEs y Framework. Será capaz además de intercambiar, conectar y diseñar artefactos de software asumiendo la obligación de analizar críticamente, interpretar y mejorar lo recibido, así como también desarrollar proyectos grupales durante los cuales se simulan condiciones similares a las del trabajo profesional.

Ejes de contenidos. Descriptores

Desarrollo de un proyecto en una herramienta aplicada a la actualidad de programación web. Manejo de Bases de Datos en el entorno de trabajo. Diseño para el cambio: adaptación a otras lenguas o localismos, adaptación a la diversidad de condiciones humanas. Notación para especificar interfaces. El manejo de los errores del usuario o del sistema. Principios de interfaces gráficas. Interfaces para un sistema nativo, uso del browser para sistemas que operen en la web. Certificados SSL. Websocket. Responsividad y Navegación. Frameworks para el desarrollo web. Paradigmas de programación web. Programación del lado del servidor. Principios de ingeniería web. Sitios web estructurados mediante bases de datos. Tecnologías de búsqueda en web. El papel del “middleware”, herramientas de apoyo. Aplicaciones locales: aplicaciones basadas en tecnologías para toda la empresa. Aplicaciones cooperativas. Creación, diseño, administración e integración de sitios web.

▪ **6.4.3.2 Desarrollo Móvil**

Formato: Asignatura

Régimen de cursada: Anual

Ubicación en el diseño curricular: 3er año

Asignación de horas semanales: 4 hs.

Total de horas: 128 hs.

Finalidades formativas de la Unidad Curricular

Desarrollar mediante las herramientas actuales distintos tipos de APPs para los distintos dispositivos y los sistemas de gestión de entorno de desarrollo. Crear, administrar y gestionar, con recursos avanzados de programación, distintos tipos de aplicaciones móviles, interfaces y depuración de errores. Concentrar el desarrollo para los distintos dispositivos. Aplicar conceptos de diseño y programación. Aplicar conceptos de orientación a objeto.

Ejes de contenidos. Descriptores

Desarrollo de un proyecto en una herramienta aplicada a la actualidad de programación móvil. Multiplataformas híbridas y nativas. Compilar a nativo. Apps multiplataforma sobre HTML5, CS33 y Javascript (Webapp). Apps interconectadas con el software y el hardware, permitiendo el uso del micro, la cámara o el GPS. Diseño de interacción.

Estructuras, Sintaxis, Estructuras de control de flujo, Bucles, Interfaz, Escenas, Eventos, Debug, Estilos, Constantes, Responsividad, Navegación. Almacenamiento local: Local y Session Storage. Almacenamiento local HTML5 – Web SQL. Integración con Api-Rest. Login y Autenticación, Cámara, Location y Publicación. Puesta en práctica de un IDE (Entorno de Desarrollo Integrado).

▪ **6.4.3.3 Gestión de Proyectos de Software**

Formato: Asignatura

Régimen de cursada: Anual

Ubicación en el diseño curricular: 3er año

Asignación de horas semanales: 4 hs.

Total de horas: 128 hs.

Finalidades formativas de la Unidad Curricular

Consolidar la formación para el trabajo en un equipo organizado en función del proyecto utilizando un CRM (Customer Relationship Management o Gestor de Relaciones con el Cliente), asumiendo las responsabilidades que se le asignaron al tiempo que interactúa con pares y líderes para lograr un mejor proceso conjunto.

Ejes de contenidos. Descriptores

Gestionar proyectos de software open source con CMR. Instalación CRM. Virtualización. Configuración base del CRM. Configuración de la Base de Datos. Interface de usuario. Móvil. Gestión de Datos. Gestión de Flujos. Gestión de Etapas. Gestión de actividades. Definición, planificación, verificación y control del avance. Roles y responsabilidades. Reportes: Análisis de cohorte, Análisis de Oportunidades, Análisis de leads. Integraciones: Ventas, Inventarios, APIs, Google Docs. Programas de trabajo y resolución conjunta de problemas. Modelos de ciclo de vida del software CRM y de procesos de desarrollo. El problema del mantenimiento y las migraciones de plataforma.

6.5 CAMPO DE FORMACIÓN DE LA PRÁCTICA PROFESIONALIZANTE

Finalidades formativas de las Unidades Curriculares

“Se entiende por prácticas profesionalizantes aquellas estrategias y actividades formativas que, como parte de la propuesta curricular, tienen como propósito que los estudiantes consoliden, integren y/o amplíen las capacidades y saberes que se corresponden con el perfil profesional en el que se están formando. Son organizadas y coordinadas por la institución educativa, se desarrollan dentro o fuera de tal institución y están referenciadas en situaciones de trabajo.”

“Las prácticas profesionalizantes propician una aproximación progresiva al campo ocupacional hacia el cual se orienta la formación y favorecen la integración y consolidación de los saberes a los cuales se refiere ese campo ocupacional, poniendo a los estudiantes en contacto con diferentes

situaciones y problemáticas que permitan tanto la identificación del objeto de la práctica profesional como la del conjunto de procesos técnicos, tecnológicos, científicos, culturales, sociales y jurídicos que se involucran en la diversidad de situaciones socioculturales y productivas que se relacionan con un posible desempeño profesional."

Las prácticas profesionalizantes se orientan a generar posibilidades para realizar experiencias formativas en distintos contextos y entornos de aprendizaje complementarios entre sí. En este sentido las prácticas pueden desarrollarse:

- En un ámbito exterior a la institución educativa.
- En un ámbito interior de la institución educativa.
- En ambos contextos al mismo tiempo.

Ámbito Exterior: En las prácticas profesionalizantes insertas en el mundo del trabajo, los estudiantes ejercitan y transfieren aprendizajes ya incorporados, y también aprenden nuevos contenidos o saberes propios del ejercicio profesional, que le corresponde al campo laboral específico. Asimismo se desarrollan relaciones interpersonales, horizontales y verticales propias de la organización.

Ámbito Interior: Cuando las prácticas profesionalizantes se dan en el contexto de la institución educativa, se orientan a la implementación de proyectos institucionales: productivo o tecnológico y/o actividades de extensión, respondiendo a necesidades de la comunidad. En este ámbito se destaca que los aprendizajes están encaminados por una concepción del trabajo, en tanto práctica social y cultural, en lugar de estar centrados en las particularidades de las funciones en un lugar de trabajo determinado.

Teniendo como referencia los ámbitos explicitados anteriormente, las prácticas profesionalizantes pueden implementarse mediante diferentes formatos, respetándose claramente los propósitos y objetivos planteados para su realización. En este sentido las prácticas profesionalizantes pueden estar comprendidas en:

- Actividades en espacios reales de trabajo: las mismas se desarrollan en instituciones y organismos, a través de los cuales se generan los mecanismos de articulación (convenios, actas acuerdos, etc.) que permiten la participación de los estudiantes en lugares y horarios definidos para tal fin.
- Actividades de extensión: diseñadas para satisfacer necesidades comunitarias. Las mismas podrán definirse y planificarse en función de relevamientos y demandas que se presenten en la institución, evaluándose la pertinencia de las mismas en función de los objetivos planteados.
- Proyectos productivos de servicios: están esbozadas para satisfacer demandas específicas de determinada producción de bienes o servicios, o también puede optarse por trabajar y fortalecer requerimientos propios del instituto. Éstos dispositivos constituyen una importante herramienta para vincular la educación y el trabajo, a partir de una formación que se centra en el aprender a emprender. De esta manera los estudiantes obtienen una capacitación técnica y estratégica que les amplía las posibilidades de participación futura en el mundo productivo. En este tipo de proyectos el aprender se logra a través de la producción concreta de un bien y/o servicio en el ámbito del establecimiento educativo,

donde los estudiantes tienen la posibilidad de organizarse asumiendo diferentes roles y contando con el acompañamiento y seguimiento de los docentes.

- Proyectos tecnológicos orientados a la investigación, experimentación y desarrollo de procedimientos, bienes o servicios relevantes desde el punto de vista social y que introduzcan alguna mejora respecto de los existentes.

Cualquiera sea la tipología que adopten las prácticas profesionalizantes, las mismas deben respetar las siguientes condiciones para su implementación:

- Estar planificadas desde la institución educativa, monitoreadas y evaluadas por un docente o equipo docente especialmente designado a tal fin y una autoridad educativa, con participación activa de los estudiantes en su seguimiento.
- Estar integradas al proceso global de formación.
- Desarrollar procesos de trabajo, propios de la profesión y vinculados a fases, procesos o subprocesos productivos del área ocupacional profesional.
- Favorecer la identificación de las relaciones funcionales y jerárquicas del campo profesional, cuando corresponda.
- Hacer posible la integración de capacidades profesionales significativas y facilitar desde la institución educativa su transferibilidad a las distintas situaciones y contextos.
- Disponer la puesta en juego de valores y actitudes propias del ejercicio profesional responsable.
- Propiciar la ejercitación gradual de niveles de autonomía y criterios de responsabilidad profesional.
- Viabilizar desempeños relacionados con las habilidades profesionales.

6.5.1 PRIMER AÑO

▪ 6.5.1.1 Práctica Profesionalizante I

Formato: Asignatura

Régimen de cursada: Cuatrimestral

Ubicación en el diseño curricular: 1er año 2do Cuatrimestre

Asignación de horas semanales: 4 hs.

Total de horas: 64 hs.

Finalidades formativas de la Unidad Curricular

Las Prácticas Profesionalizantes en el primer año inician a los estudiantes en ámbitos ligados al mundo laboral que comenzarán a construir su perfil de Desarrollador. Brindan un espacio de integración y aplicación de contenidos curriculares de la carrera favoreciendo el contacto con diferentes situaciones y problemáticas que permitan la identificación del objeto de la práctica

profesional como así también del conjunto de procesos técnicos, tecnológicos, científicos, culturales, sociales y jurídicos que se involucran en la diversidad de situaciones socioculturales y productivas que se relacionan con un posible desempeño profesional. Metodología Scrum.

Ejes de contenidos. Descriptores

Reconocimiento del ámbito de trabajo, análisis de las tareas del equipo de desarrollo de artefactos de software, dentro de la organización o ámbito donde se realice esta práctica profesionalizante. Reconocer los procedimientos y etapas del desarrollo de artefactos de software.

Definir cuáles son las conductas de trabajo en una empresa, la gestión de procesos y equipos para abordar proyectos complejos que requieren entornos dinámicos y cambiantes. Metodología de trabajo ágil que tiene varios objetivos: Capacidad para colaborar, participar y aprender en su asistencia a las áreas operativas y para el trabajo en equipo tomando contacto con la operatoria y forma de organización del trabajo en distintas áreas de la empresa.

Metodologías ágiles para gestionar: kanban, scrum. Metodologías tradicionales y ágiles. Fraccionamiento del producto en unidades realizables en un scrum, cola de pendientes, mejora de productos provisionales, variación de los roles y la documentación en el marco de un proceso en el que se aplican metodologías ágiles. Elementos de administración, de la configuración y control de versiones de software. Herramientas de versionado. Otras herramientas (bibliotecas, diccionarios, repositorios) del proyecto. El proceso personal de software, estadísticas personales y capitalización de experiencias.

Manejo de las normativas de seguridad fundamentales para el futuro técnico.

Evaluación de los resultados y de su participación en todos estos procesos durante la práctica, elaboración de un trabajo de campo.

Aspectos Metodológicos

A partir de situaciones desarrolladas en el mismo ámbito de formación académica, aplicando estrategias de enseñanza que incorporan el análisis de casos, la resolución de problemas y el desarrollo de entrenamientos específicos en contextos de simulación. Se referirán a problemas y preguntas que le permitan al/la estudiante profundizar el desarrollo de las competencias necesarias y suficientes para resolver cualquier problema específico que se relacione con sus competencias profesionales. Asistencia a empresas, organismos estatales o privados o en organizaciones no gubernamentales; proyectos productivos articulados entre la institución educativa y otras instituciones o entidades bajo la supervisión y coordinación de un profesor.

6.5.2 SEGUNDO AÑO

▪ 6.5.2.1 Práctica Profesionalizante II

Formato: Asignatura

Régimen de cursada: Cuatrimestral

Ubicación en el diseño curricular: 2do año 2do Cuatrimestre

Asignación de horas semanales: 4 hs.

Total de horas: 64 hs.

Finalidades formativas de la Unidad Curricular

Las Prácticas Profesionalizantes en el segundo año serán la continuidad de lo comenzado en el primer año. Proponen un espacio de integración y aplicación de contenidos curriculares de la carrera afianzando aquellos aprendidos con los nuevos adquiridos, prestando importancia a los aspectos relevantes del desarrollo de un software en equipo y en scrum.

Ejes de contenidos. Descriptores

Reconocimiento del ámbito de trabajo, análisis de las tareas en un equipo de desarrollo de software: Product Owner, Scrum Master, Scrum Team. Herramientas digitales para desarrollar y gestionar los proyectos con una metodología Scrum: Denotar conocimientos teóricos de los elementos fundamentales de los procesos, de la planificación, del diseño y construcción de artefactos de software. Capacidad para colaborar, participar y aprender en su asistencia a las áreas operativas y para el trabajo en equipo.

Manejo de las normativas que determinan la seguridad en base de datos y sistemas fundamentales para el futuro técnico.

Evaluación de los resultados y de su participación en todos estos procesos durante la práctica, elaboración de un trabajo de campo sobre los procesos desarrollados en un equipo de desarrollo.

Aspectos Metodológicos

Las prácticas se realizarán a partir de situaciones reales en organizaciones que tengan que ver con el desarrollo de software, bajo la coordinación y supervisión de un profesor, aplicando estrategias de enseñanza que incorporan el análisis de casos, la resolución de problemas y el desarrollo de entrenamientos específicos en contextos de simulación. Se referirán a problemas y preguntas que le permitan al alumno profundizar el desarrollo de las competencias necesarias y suficientes para resolver cualquier diagnóstico específico que se relacione con sus competencias profesionales. Las diferentes situaciones que pueda observar y/o participar, y/o desarrollar, le permitirá al estudiante ir consolidando la formación y su perfil profesional, de manera tal que les facilite su inserción laboral contando con esta experiencia previa.

6.5.3 TERCER AÑO

▪ 6.5.3.1 Práctica Profesionalizante III

Formato: Asignatura

Régimen de cursada: Anual

Ubicación en el diseño curricular: 3er año

Asignación de horas semanales: (según organización institucional)

Total de horas: 128 hs.

Finalidades formativas de la Unidad Curricular

En las prácticas profesionalizantes del último año será fundamental para que el/la estudiante pueda aplicar los diversos criterios y técnicas de un proceso integral de desarrollo de software full stack, con el fin de elaborar un trabajo final de carrera para un cliente real que satisfaga las necesidades específicas. Si se realiza utilizando la metodología Scrum: Definir los roles y funciones de los integrantes del scrum, código aportado, su función, backend-frontend.

Ejes de contenidos. Descriptores

Desarrollo de artefactos de software integrados móvil-web: backend y frontend o CRM, para una organización o institución pública o privada. Análisis de las dimensiones a tener en cuenta para desarrollar la práctica de su profesión, con estudio de las necesidades del cliente.

Accionar con los diferentes actores de la organización pudiendo participar de las distintas etapas del proceso de un proyecto de desarrollo de software.

Desarrollar su propio proyecto final donde el proceso incorpore las etapas, la planificación, el diseño y la construcción de los artefactos de software; la comunicación de la propuesta y la puesta en práctica.

Trabajo de campo con clientes reales y problemáticas reales donde sintetice las aptitudes y los conocimientos aportados por el área durante la carrera.

Aspectos Metodológicos

Las prácticas se realizarán en una organización o institución pública o privada, bajo la coordinación y supervisión de un/a profesor/a quien realizará el acompañamiento durante el proceso del análisis del caso, la formulación de un proyecto y su desarrollo. Durante el proceso trabajarán con problemas y preguntas que le permitan al/la estudiantado profundizar el desarrollo de las competencias necesarias y suficientes para resolver cualquier problemática específica que se relacione con sus competencias profesionales, logrando así su inserción laboral contando con esta experiencia previa.

Elaboración de un informe final de la práctica de trabajo con una propuesta de desarrollo personal.

CAPÍTULO VII

BIBLIOGRAFÍA GENERAL

- Bunge, M. (2005) La Ciencia, su Método y su Filosofía. Buenos Aires. Ed. De bolsillo Colección Ensayo Ciencia.
- Camilloni, A. y otros (2007) El Saber Didáctico. Buenos Aires. Ed. Paidós.
- Cotlar, M. y de Sadosky, C.R. (1977) Introducción al Álgebra. EUDEBA.
- Cramer, H. (1998) Elementos de la Teoría de Probabilidades y Aplicaciones. Aguilar.
- Díaz Barriga, Á. (2009) Pensar la Didáctica. Buenos Aires. Ed. Amorrortu.
- Díaz Barriga, F. (1990) Metodología de Diseño Curricular para Educación Superior. México. Ed. Trillas.
- GRADY, Booch Análisis y Diseño Orientado a Objetos con Aplicaciones. Addison-Wesley.
- Joyanes, A. L. (2004) Algoritmos y Estructuras de Datos una Perspectiva en C. Madrid. Ed. McGraw-Hill.
- CAÑOS, José H.; LETELIER, Patricio; PENADÉS, Ma Carmen; Metodologías Ágiles en el Desarrollo de Software.
- Joyanes, A. L. (2008) Fundamentos de la Programación. Madrid. Ed. Mc Graw-Hill.
- BATTINI, CERI, NAVATHE. Diseño Conceptual de Bases de Datos. Mc Graw Hill.
- RAMAKRISHNAN RAHGU. Sistemas de Gestión de Bases de Datos Mc Graw-Hill.
- Quiroga, P. (2010) Arquitectura de Computadoras. Buenos Aires. Ed. Alfaomega Grupo Editor.
- S. Krug, (2014), 'Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability', New Riders
- S. M. Weinschenk, (2016) '100 cosas MÁS sobre la gente que cada diseñador necesita saber', Anaya.
- Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. (1998) Design Patterns. Elements of Reusable Object-Oriented Software. Ed. Addison – Wesley.
- Glendinning, E. y Mc Ewan, J. (2004) Basic English for Computing. Oxford, Estados Unidos. New Edition.
- Repetto, C. (1989). Manual de Análisis Matemático. Argentina, Ed. Macchi.
- Rabuffetti, H. T. (1984). Introducción al Análisis Matemático. Buenos Aires. El Ateneo.
- Enrique Amodeo, (2013) Principios de diseño de APIs REST. Leanpub.
- L. Richardson, M. Amundsen y S. Ruby (2013) RESTful Web APIs: Services for a Changing World. O'Reilly Media.
- Welling, L. y Thomson, L. (2005). Desarrollo Web con PHP y MYSQL. Ed. Anaya Multimedia, México.
- Tanenbaum, A. (1999). Sistemas Operativos, Diseño e Implementación. México. Ed. Prentice Hall.
- Tanenbaum, A. S. (2009). Sistemas Operativos Modernos. España. Ed. Prentice Hall.
- Stallings, W. (2003). Fundamentos de Seguridad en Redes. México. Pearson Educación.
- Stallings, W. (2004). Comunicaciones y Redes de Computadores. España. Ed. Pearson Education.
- Guariglia, O. (2002). Una Ética para el Siglo XXI. Capítulo 2 y 3. Buenos Aires. Ed. Fondo de Cultura Económica.
- Huidobro Moya, J. y Roldan Martínez, D. (2005). Seguridad en Redes y Sistemas Informáticos. Madrid. España. Thomson Paraninfo.
- ISO/IEC 27001: 2005, (2005). Sistemas de Gestión de la Seguridad de la Información – Requisitos.

Organización Internacional de Normalización. Ginebra. Suiza.
ISO/IEC 27002: 2005, (2005). Código de Buenas Prácticas para la Gestión de la Seguridad de la Información. Organización Internacional de Normalización. Ginebra. Suiza.
Kendall, J. y Kendall, K. (2005). Análisis y Diseño de Sistemas. México. Ed. Pearson Education.
Fernando Luna, (2019) PROGRAMADOR WEB FULL STACK - Desarrollo frontend y backend: Ecosistema Web. Creative Andina Corp.
Eric Bush, (2016). Full-Stack JavaScript Development. Red Sky.
Mario Rubiales Gómez, (2017) Curso de Desarrollo Web: HTML, CSS y JavaScript. Edición 2018. Anaya Multimedia.
Arturo Montejó Ráez y Salud María Jiménez Zafra. (2019). Curso de Programación Python. Anaya. Multimedia.

Manuales:

Diccionario Oxford Básico Español Inglés-Inglés Español. Editorial Oxford University Press.
Manual de Programación Web. Computer Aided Education S.A.
Manual del sistema operativo Linux.
Manual diseño de páginas Web. Sistemas Operativos –Conceptos y Diseño. Computer Aided Education S.A.

CAPÍTULO VIII

GLOSARIO TÉCNICO – DESARROLLO FULL STACK

A

Abstraction: propiedad y/o técnica de software que oculta los detalles de la implementación.

AJAX: técnica de desarrollo web para crear aplicaciones web asíncronas.

Algoritmo: conjunto de instrucciones concretas y detalladas mediante el cual se consigue una acción determinada. Los algoritmos ayudan al programador a planificar un programa antes de su escritura en un lenguaje de programación.

Ancho de banda: es la cantidad de información que puede intercambiarse mediante una conexión de red en un período dado.

Android: sistema operativo dedicado principalmente a los dispositivos móviles.

Apache: servidor web de distribución libre.

API: Interfaz de Aplicación del Programa o Interfaz de programación de aplicaciones es un conjunto de definiciones, protocolos, esquemas, herramientas y prácticas para comunicarse con la aplicación.

Aplicación o APP: programa que se instala en un dispositivo móvil y que se puede integrar a las características del gadget, como su cámara o sistema de posicionamiento global. Además se puede actualizar para añadirle nuevas características con el paso del tiempo.

Applet: componente de una aplicación que se ejecuta en el contexto de otro programa, por ejemplo, en un navegador web.

Array: colección finita de datos del mismo tipo, que se almacenan en posiciones consecutivas de memoria y reciben un nombre común.

Arquitectura de sistemas: descripción del diseño y estructura de un sistema informático, tanto desde el punto vista hardware como software.

Arquitectura de software: un conjunto de patrones que definen la estructura de un sistema software, es decir, los elementos que componen el sistema, sus propiedades y sus relaciones.

ASCII: American Standard Code of InformationInterchange: Código normalizado estadounidense para el intercambio de la información, permite definir caracteres alfanuméricos y se lo usa para lograr compatibilidad entre diversos procesadores de texto.

B

Backup: copia de seguridad. Se hace para prevenir una posible pérdida de información.

backend: es la parte que procesa la entrada desde el frontend. La separación del sistema en frontends y backends es un tipo de abstracción que ayuda a mantener las diferentes partes del sistema separadas.

Base de datos: colección de datos organizados y almacenados para una fácil recuperación utilizada por el programa informático, de manera que pueda ser gestionada fácilmente.

Base de datos relacional: es una base de datos organizada con el modelo relacional.

Base de datos NoSQL: base de datos no relacional o no SQL. A diferencia de las bases de datos relacionales, utiliza otras formas que no son datos tabulares, como colecciones de valores clave, estructuras multinivel, gráficos, etc.

Binario: representación numérica en base 2.

Biblioteca: colección de funciones o rutinas predefinidas que un programa puede utilizar.

Bloque: sentencias y declaraciones encerradas entre una pareja de llaves. Un bloque delimita un nivel de ámbito.

Bootstrap: framework frontend utilizado para desarrollar aplicaciones web y sitios móviles.

Bug: error en el código de un programa.

C

C: lenguaje orientado a la implementación de sistemas operativos, concretamente Unix.

Clase: una plantilla para la creación de objetos de datos según un modelo predefinido, una colección encapsulada de datos y operaciones que actúan sobre los datos, fundamental en programación orientada a objetos.

Cliente/Servidor: arquitectura de sistemas que distingue dos partes: el servidor, encargado de ofrecer servicios tales como datos, documentos, aplicaciones y el cliente, que es quien realiza peticiones al servidor para obtener dichos servicios.

CMS: sistema de gestión de contenidos para páginas web.

Código: son las instrucciones que escribe un programador en un lenguaje de programación.

Código ejecutable o código objeto: código generado por un compilador o un ensamblador a partir de un código fuente de un programa y que generalmente puede ser ejecutado directamente por el procesador de un sistema informático.

Código fuente: texto que contiene las instrucciones del programa, escritas en el lenguaje de programación.

Compilación: proceso de traducción de un lenguaje de programación.

Compilador: programa de software que realiza un proceso de compilación (traducción del lenguaje fuente a lenguaje máquina) de un programa escrito en un lenguaje de programación de alto nivel.

Contenedor: clase que implementa una estructura de datos que contiene una colección de objetos.

Controlador: o driver, es una aplicación informática integrada en un sistema operativo que permite interactuar con un dispositivo interno o externo al sistema informático.

CSS: (Hojas de estilo en cascada) es un método de asignar reglas de formato a una página HTML que permite separar el contenido y la presentación de un sitio web.

D

Data modeling: proceso de documentar el diseño de un sistema de software complejo como un diagrama fácil de entender, utilizando texto y símbolos para representar la forma en que los datos deben fluir.

Depurador: programa diseñado para ayudar a programar sin errores una aplicación, ayudando a identificar el error a corregir.

Desarrollador backend: crea todos los procesos de código de una página que el usuario no puede ver. Maneja la base de datos y la lógica de una página web, está involucrado en el proceso de combinar un servidor, una aplicación y una base de datos para resolver un problema (entorno del lado del servidor). Todo este proceso está sólidamente arraigado en la lógica, una red de procesos y consultas que se resuelven en fracciones de segundo para darle un cierto resultado deseado como usuario. Se diferencian de los desarrolladores de frontend en que el trabajo de un desarrollador de backend está completamente libre de cualquier diseño visual. En cambio, se basa en el razonamiento lógico y la arquitectura de software que tiene por objeto ofrecer un resultado determinado.

Desarrollador frontend: hace todo lo que el usuario ve en la página. Le da el diseño a las páginas web a través de herramientas como HTML y CSS. Con estas herramientas ellos pueden modificar el color de fondo, los tipos de fuente, el color de las letras o la estructura de texto en una página. También hacen que las páginas sean dinámicas, o sea que el usuario pueda interactuar con el sitio web.

Desarrollador Full Stack: Los desarrolladores Full Stack son perfiles híbridos que conocen las herramientas que usa el programador frontend y backend. Pueden hacer desde el diseño de una página web hasta manejar la base de datos y procesos de la página.

Desarrollador Móvil: Los desarrolladores móviles crean APPs para celulares o tablets.

Diseñador UI/UX: UI User Interface (o Interfaz de usuario), es lo que el usuario puede ver en una aplicación. Mientras que UX es User Experience (o Experiencia de Usuario) y es lo que el usuario siente cuando usa una aplicación. Un Diseñador UI/UX diseña cómo se va a ver la página web o app en todos los dispositivos como computadora, celular y tablet. Se encarga de crear una buena experiencia para el usuario a través del diseño de la aplicación.

Desarrollo ágil de software: se refiere a métodos y prácticas de **desarrollo de software** que brindan soluciones a través de la colaboración entre equipos auto organizados y multifuncionales. Incluye: planificación, [análisis de requisitos](#), diseño, codificación, pruebas y documentación. Los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto.

Desarrollador de la pila completa: es alguien que se siente cómodo trabajando con tecnologías tanto de fondo como de frente.

Despliegue: significa empujar la nueva versión del paquete de software a su entorno de destino.

Diseño: actividad de definir cómo se debe estructurar e implementar un programa, consiste en la definición de los recursos software que van a ser utilizados para desarrollar una aplicación conforme a los requisitos indicados en unas especificaciones dadas.

Diseño orientado a objetos OOD (object.oriented design): Diseño realizado en términos de objetos, clases y selecciones de clases.

Diseñador de la interfaz de usuario: es el que diseña cómo se ve la aplicación desde la perspectiva del usuario para permitirles que interactúen con la aplicación.

Diseñador de UX o "Experiencia del usuario": ayuda a crear una mejor experiencia de uso de la aplicación. Su trabajo es hacer la aplicación tan fácil y útil para los usuarios como sea posible para aumentar la usabilidad.

Dispositivo de Almacenamiento: componente hardware que se utiliza para grabar y leer datos.

Docker: es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos.

Dominio: Básicamente es el texto que ingresa una persona en un navegador para ingresar a un sitio web en específico. Por ejemplo el dominio del Instituto es "itscipolletti.edu.ar".

E

Encapsulamiento, encapsulación: localización y protección de las características internas y estructura de un objeto. Combinación de métodos y datos en una única estructura de datos.

ERP: Enterprise Resource Planning, se refiere al tipo de software que usan las organizaciones para administrar sus actividades.

Etiqueta: (tag) Una instrucción HTML que indica a un navegador Web como visualizar un documento.

F

Framework: es un entorno de trabajo que tiene como objetivo facilitar la labor de programación ofreciendo una serie de características y funciones que aceleran el proceso, reducen los errores, favorecen el trabajo colaborativo y consiguen obtener un producto de mayor calidad.

Freelancer: es la persona que puede trabajar desde cualquier parte del mundo con una computadora y una conexión a internet, se encargan de desarrollar o programar software, complementos de software, páginas web, o cualquier otra cosa que involucre códigos.

Frontend: es la parte del software que interactúa con los usuarios.

Full Stack: es el profesional que domina las dos partes de un desarrollo web; tanto los aspectos del frontend, la parte visual con la que el usuario puede interactuar, como los del backend, la gestión interna de la página y de la comunicación entre el navegador y la base de datos.

G

Git: es un sistema de control de versiones distribuido de código abierto y gratuito para rastrear los cambios en los archivos de computadora y coordinar el trabajo sobre esos archivos entre varias personas.

GitHub: es un popular servicio de alojamiento de Git. Se utiliza principalmente para el código, proporciona varias características de colaboración, términos importantes de TI como seguimiento de errores, solicitudes de características, gestión de tareas y wikis.

GitLab: es un repositorio de alojamiento de Git que proporciona wikis y funcionalidad de seguimiento de problemas.

GNU-Linux: es la denominación técnica y generalizada que reciben una serie de sistemas operativos de tipo Unix, que también suelen ser de código abierto, multiplataforma, multiusuario y multitarea.

H

Hardware: conjunto de componentes materiales de un sistema informático o digital.

HTML (Hypertext Markup Language): Lenguaje de 'script' o de marcas para diseñar páginas Web.

I

IDE o Entorno de Desarrollo Integrado: es una aplicación de edición de código diseñada para ayudar a los programadores a escribir, ejecutar y depurar el código.

Interfaz: en programación orientada a objetos, una interfaz es un medio común para que los objetos no relacionados se comuniquen entre sí.

Intérprete: es un programa informático capaz de analizar y ejecutar otros programas.

J

Java: es un lenguaje de programación compilado, orientado a objetos, similar en sintaxis a C++. Su intención es permitir a los desarrolladores de aplicaciones "escribir una vez, ejecutar en cualquier lugar.

JavaScript: también conocido como JS es un lenguaje dinámico de alto nivel, basado en prototipos, sin teclar. Dependiendo del entorno, JavaScript puede ser interpretado o compilado. Es un lenguaje de programación multiparadigma que soporta estilos de programación orientados a objetos, imperativos y funcionales.

Jerarquía de clases: Colección de clases organizadas en términos de relaciones de superclases y subclases.

JVM Máquina Virtual Java: una emulación de software de una máquina que puede ejecutar bytecodes de Java. Proporciona una implementación del procesador, sistema de memoria e interfaces a dispositivos hardware.

K

Kanban: es un método visual que se utiliza para controlar las tareas a través de su división por fases hasta su finalización.

L

LAMP: acrónimo compuesto por las iniciales de sus cuatro componentes: Linux, Apache, MySQL y PHP forman la infraestructura en el servidor que hace posible la creación y el alojamiento de páginas web dinámicas.

Lenguaje de consulta: SQL es un lenguaje informático usado para hacer consultas en bases de datos y sistemas de información, recuperación de información.

Lenguaje de programación: es un lenguaje artificial que puede ser interpretado por un sistema informático mediante un programa compilador o un intérprete para su traducción a instrucciones del procesador (lenguaje máquina), el cual se utiliza para el desarrollo de software.

Lenguaje estructurado: lenguaje de programación diseñado según el enfoque de programación estructurada.

Lenguaje de manipulación de datos: en el contexto de las bases de datos, un lenguaje de manipulación de datos es aquel específicamente diseñado para las funciones de manipulación de datos tales como la inserción, modificación y borrado de datos. Se utiliza en bases de datos de almacenamiento estructurado.

Librerías (bibliotecas) de software: conjunto de programas agrupados que pueden ser utilizados por otros programas para el desarrollo de software.

Log: es un registro de lo que ha pasado.

M

MER-DER: es un modelo entidad-relación, una herramienta para el modelo de datos que facilita la representación de entidades de una base de datos, permite valorar cómo es de buena una señal modulada digital y un tipo de modelo de datos conceptual de alto nivel que se emplea en el diseño de las bases de datos relacionales.

Método abstracto: método que sólo tiene signature y no tiene cuerpo y debe estar contenido dentro de una clase abstracta.

Método de la clase: sinónimo de método estático. Un método que se puede invocar sin crear una instancia de la clase.

Método de la instancia: procedimiento declarado por una clase que se llama por sus objetos de instancias (o los de las subclases).

Modelo de OSI: modelo de interconexión de sistemas abiertos conocido como "modelo OSI", es un modelo de referencia para los protocolos de la red creado en el año 1980 por la Organización Internacional de Normalización. Describe una estructura con siete capas para las actividades de red. Cada capa tiene asociados uno o más protocolos. Las capas representan las operaciones de transferencia de datos comunes a todos los tipos de transferencias de datos entre las redes de cooperación.

Moldeado: proceso de convertir un valor de un tipo de dato primitivo en otro tipo primitivo o conversión de un objeto de un tipo de dato en otro tipo de objeto.

N

Networking definido por Software: es la posibilidad que ofrecen los dispositivos y los distintos sistemas de programar la infraestructura completa de red.

O

Oauth: protocolo que permite autorización segura de una API de modo estándar y simple para aplicaciones de escritorio, móviles y web.

Ocultamiento: hace referencia a que los atributos privados de un objeto no pueden ser modificados ni obtenidos a no ser que se haga a través del paso de un mensaje.

Objeto: es una unidad dentro de un programa informático que tiene un estado y un comportamiento

Open Source: código fuente abierto software libre, se refiere a un programa cuyo código fuente está disponible al público en general, de forma gratuita, para usar y modificar.

Operaciones CRUD: resume las funciones requeridas por un usuario para crear y gestionar datos. Estas operaciones están específicamente adaptadas a los requisitos del sistema y de usuario, ya sea para la gestión de bases de datos o para el uso de aplicaciones.

Orientación a objetos: es una forma de enfocar el desarrollo de software basada en la identificación de los datos y las operaciones encapsulados en una entidad denominada objeto.

P

Paquete: colección de clases agrupadas juntas.

Paquete de instalación: agrupación de recursos software (programas, librerías, ficheros de configuración, datos, etc.) empaquetados generalmente en un único fichero comprimido para su distribución y despliegue.

Parámetro: variable utilizada para recibir valores de entrada en una rutina, subrutina o método.

Periféricos: dispositivos hardware independientes conectados a un sistema informático.

PHP: es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web.

Polling: es una forma de control en redes de área local.

Proceso: es un programa que entra en ejecución.

Programa: conjunto de instrucciones o sentencias que describen alguna aplicación o actividad ejecutada en una computadora.

Programación: proceso de desarrollar programas informáticos: crear, probar, depurar y mantener el código fuente de las aplicaciones informáticas. Generalmente implica el diseño de algoritmos y estructuras de datos y la codificación en uno o varios lenguajes de programación.

Programación controlada por sucesos: los códigos se ejecutan por activación de sucesos, tales como pulsar un botón o mover el ratón.

Programación estructurada: es un estilo de programación que utiliza estructuras sencillas tales como secuencias, condicionales, bucles, etc.

Programación imperativa: basada en los principios de instrucción o secuencias de órdenes, selección, repetición, variables y asignación.

Programación orientada a objetos OOP (objectoriented programming): modelo de programación informática que organiza el diseño de software en torno a objetos, en lugar de funciones y lógica. Es un paradigma de programación que viene a innovar la forma de obtener resultados. Los objetos manipulan los datos de entrada para la obtención de datos de salida específicos, donde cada objeto ofrece una funcionalidad especial.

Protocolo (de comunicaciones): conjunto de reglas y normas que permiten que dos o más nodos de una red se comuniquen entre ellos para transmitir y recibir información.

Prueba de software: proceso que consiste en la comprobación del software desarrollado en la fase de implementación no contiene errores.

Python: es uno de los lenguajes de programación más utilizados.

R

R: es un lenguaje de programación de código abierto y un entorno de software para la computación estadística y los gráficos.

Red: Infraestructura que permite a los ordenadores comunicarse unos con otros.

Red de Área Local LAN: (Local Area Network) es la interconexión de varias computadoras y periféricos entre sí. Su extensión está limitada a un entorno cercano.

Rest API: es una interfaz de programación de aplicaciones, ideal para el Internet de las cosas y el desarrollo de aplicaciones para dispositivos móviles.

Responsive: filosofía de diseño y desarrollo web que indica que una página o aplicación web se debe de adaptar al dispositivo que usa el usuario, de tal manera que no importa si estás viendo una páginas web en notebook, PC, tablet o smartphone, en todos se ve bien.

Reusabilidad: posibilidad de que un elemento de software pueda ser usado nuevamente sin necesidad de ser reimplementado, simplificando y agilizando el desarrollo de aplicaciones.

Ruby: es un lenguaje de programación dinámico, reflexivo, orientado a objetos y de propósito general.

S

Servidor: Es una computadora que, formando parte de una red, provee servicios a otras computadoras.

Scripting: tipo de lenguaje que está diseñado para integrarse y comunicarse con otros lenguajes de programación.

Scrum: marco de trabajo para desarrollo ágil de software en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente en equipo para obtener el mejor resultado posible de proyectos.

Servidor LDAP: basado en la conexión entre cliente y servidor, almacena los datos relativos al directorio.

Servidores web: es un software que forma parte del servidor y tiene como misión principal devolver información cuando recibe peticiones por parte de los usuarios, les permite ver una página web en su navegador cuando quieren hacerlo.

Shell: es una interfaz que permite al usuario interactuar con una computadora.

Sistema de control de versiones: es la gestión de los cambios en los documentos, programas informáticos, grandes sitios web y otras colecciones de información.

Sistema operativo: conjunto de programas que controlan el funcionamiento de una computadora, es el encargado de gestionar los elementos de un sistema informático (procesos, memoria, almacenamiento, etc.) así como los periféricos conectados a él.

Sistemas de ficheros: son los componentes que estructuran la información guardada en una unidad de almacenamiento.

SOAP Simple Object Access Protocol, Protocolo de Acceso de Objeto Simple, es una manera de intercambiar información estructurada entre computadoras.

Software: conjunto de componentes no materiales de un sistema informático o digital.

Sockets: método para la comunicación entre un programa del cliente y un programa del servidor en una red, se define como el punto final en una conexión.

SQL: lenguaje de programación de bases de datos.

SSL/TLS: SSL Secure Sockets Layer (Protocolo de Capa de Conexión Segura) y TSL Transport Layer Security (Seguridad de la Capa de Transporte), son protocolos de permiten una comunicación segura entre los nodos de una red, generalmente pública como Internet, utilizando mecanismos criptográficos. Proporcionan integridad, privacidad y autenticación de los sistemas informáticos intervinientes en la comunicación.

Socket: término que describe la facilitación de comunicación entre un servidor y un cliente.

T

TCP/IP: protocolo de Internet que toma su nombre de los dos protocolos más importantes: TCP (Protocolo de control de la transmisión) e IP (Protocolo de Internet).

Tipos de datos: atributo asociado a un dato o variable que indica sus características y que facilita al compilador la codificación en memoria del mismo.

U

UML: (lenguaje unificado de modelización) es lenguaje gráfico y su uso se basa en la creación de diferentes diagramas que abarcan diferentes aspectos de un sistema software: estructura, actividades, estados, etc.

Unicode: es el estándar de codificación de caracteres universal utilizado para la representación de texto para procesamiento del equipo.

Usuario: individuo que utiliza una computadora, sistema operativo, servicio de red o cualquier sistema. También se usa USUARIO para clasificar a diferentes privilegios, permisos a los que tiene acceso para interactuar o ejecutar.

