



Gobierno de la Provincia de Mendoza
República Argentina

Resolución

Número:

Mendoza,

Referencia: DGE - Apruébese el plan de estudio “Tecnicatura Superior en Desarrollo de Software”

VISTO, el EX-2024-08748408- -GDEMZA-MESA#DGE, caratulado “Tecnicatura Superior en Desarrollo de Software”, y;

CONSIDERANDO:

Que la oferta educativa propuesta se desarrolla respetando el marco establecido por la Ley Nacional de Educación N° 26.206, la Ley de Educación Superior N° 24.521, la Ley de Educación Técnico - Profesional N° 26.058, la Ley Provincial de Educación N° 6.970, la Resolución N° 047-CFE- 08, y sus modificatorias Resoluciones N° 209-CFE-13, 229-CFE-14 y 295-CFE-16 que establecen los lineamientos para la organización institucional y curricular de la Educación Técnico Profesional, la Resolución N° 1485-DGE-17 y el Decreto N° 530/18;

Que la Educación Superior tiene por finalidad proporcionar formación científica, profesional, humanística y técnica en el más alto nivel y atender tanto a las expectativas y demandas de la población como a los requerimientos del sistema cultural y de la estructura productiva;

Que la Educación Técnico Profesional es parte integrante y sustantiva del Sistema Educativo Nacional y constituye una herramienta estratégica para el desarrollo económico, social, cultural y político de la Nación;

Que las propuestas de nuevas ofertas de nivel superior vinculadas a la formación técnico-profesional procuran introducir a los estudiantes en una trayectoria de profesionalización garantizando su acceso a una base de conocimientos, habilidades, destrezas, valores y actitudes profesionales que le permitan el ingreso al mundo de los saberes y del trabajo dentro de un campo profesional determinado;

Que las competencias profesionales permitirán colaborar con la integración y participación de los distintos actores locales para el desarrollo territorial a escala regional;

Que la titulación que otorga una carrera de Nivel Superior debe responder a una demanda diferenciada de formación de recursos humanos calificados, en estrecha relación con necesidades socio productivas y culturales, que puedan insertarse eficientemente en el mundo del trabajo;

Que esta formación se orienta a un nivel profesional que le permite al egresado enfrentar problemas cuya resolución implica el conocimiento de los principios científicos tecnológicos, éticos y socioculturales involucrados en su área;

Que actualmente las exigencias del mundo productivo plantean la necesidad de definir un modelo de desarrollo regional y rural, no solamente a partir de políticas económicas sino también educativas;

Que desde la Dirección de Educación Superior, y con la participación de Instituciones Educativas de Nivel Superior y representantes especialistas del sector de la salud de la provincia se constituyó una mesa de trabajo para la homologación curricular de la carrera;

Que la propuesta definitiva ha sido elevada al Honorable Consejo Administrativo de la Enseñanza Pública;

Que en orden 4 rola visto bueno de la Subsecretaria de Educación;

Que la carrera constituye un proyecto de calidad, de acuerdo con las competencias propuestas, se ajusta a la normativa vigente del nivel y cuenta con informe del Ministerio de Salud de la Provincia.

Por ello,

EL DIRECTOR GENERAL DE ESCUELAS

RESUELVE:

Artículo 1ro.- Apruébese el plan de estudio que homologa la carrera de “**Tecnicatura Superior en Desarrollo de Software**”, cuyo diseño curricular forma parte integrante de la presente Resolución, como ANEXO (archivo embebido).

Artículo 2do.- Deróguense la Resolución N° 501-DGE-19, en todos sus términos.

Artículo 3ro.- Determinése que el plan de estudio correspondiente a la carrera “**Tecnicatura Superior en Desarrollo de Software**”, previo a ser implementado en las instituciones de nivel superior de gestión estatal y privada, deberá contar con una norma específica de la Dirección de Educación Superior que autorice la oferta educativa y la posterior matriculación de alumnos.

Artículo 4to.- Determinése que para la emisión de la norma que autoriza la matriculación de estudiantes en institutos de gestión privada se deberá contar, además, con la previa autorización de la Dirección de Educación Privada.

Artículo 5to.- Tramítese, por intermedio de la Dirección de Educación Superior, el reconocimiento de la Validez Nacional de la carrera aprobada, en acuerdo con lo determinado por el Ministerio de Educación de la Nación y el Consejo Federal de Educación.

Artículo 6to.- Comuníquese a quienes corresponda.

Digitally signed by GARCIA ZALAZAR Manuel Tadeo
Date: 2024.11.26 08:31:01 ART
Location: Ciudad Autónoma de Buenos Aires

Digitally signed by GDE GDEMZA - Gestion Documental Electronica
MENDOZA
DN: cn=GDE GDEMZA - Gestion Documental Electronica
MENDOZA, c=AR, o=Ministerio de Gobierno Trabajo y Justicia,
ou=Direccion General de Informatica y Comunicaciones,
serialNumber=CUIT 30999130638
Date: 2024.11.26 08:31:02 -0300'

ANEXO

I. ESPECIFICACIÓN DE LA CARRERA

- **NOMBRE DE LA CARRERA:** *Tecnicatura Superior en Desarrollo de Software*
- **TÍTULO QUE OTORGA:** *Técnico/a Superior en Desarrollo de Software*
- **FAMILIA PROFESIONAL:** *Informática*
- **SUBSECTOR:** Software y Servicios Informáticos
- **CARGA HORARIA:** *2790 horas cátedra / 1860 horas reloj.*
- **MODALIDAD:** *Presencial / Bimodal*
- **FORMATO DE LA CARRERA:** *Disciplinar*
- **DURACIÓN:** *6 cuatrimestres*
- **CARÁCTER:** Diversificada y/o Especializada para las trayectorias profesionales previas de Informática o similares.
- **CONDICIONES DE INGRESO:**

Haber aprobado el Nivel Secundario o Ciclo Polimodal, o bien, ser mayor de 25 años según lo establecido en el Art. 7° de la Ley de Educación Superior N° 24.521 y cumplimentar lo establecido en la normativa provincial vigente.

II. PERFIL PROFESIONAL:

Competencias

Las competencias de un Desarrollador de Software pueden variar dependiendo del contexto y el área específica en la que se desempeñe, sin embargo, algunas de las competencias más comunes incluyen:

- **Conocimientos técnicos:** los lenguajes de programación, infraestructura y tecnologías relevantes para el puesto de trabajo.
- **Habilidades de resolución de problemas:** investigar, identificar y resolver problemas de manera efectiva.
- **Habilidades de programación:** producir código eficiente, claro y mantenible.
- **Habilidades de comunicación:** comunicarse de manera clara y efectiva tanto con los equipos de trabajo como con los clientes.
- **Habilidades de colaboración:** trabajan en equipo, por lo que es importante que sean capaces de colaborar y trabajar de manera efectiva con otros miembros del equipo.
- **Habilidades de aprendizaje continuo:** que estén dispuestos a aprender y a mantenerse actualizados con las últimas tendencias y tecnologías.
- **Conocimientos de metodologías ágiles:** trabajar y gestionar en forma colaborativa con distintos equipos de trabajo actividades y requerimientos.

Competencias Profesionales

1. *Analizar requerimientos y diseñar soluciones de software, comprendiendo las especificaciones funcionales, no funcionales e infraestructura tecnológica.*
2. *Desarrollar software utilizando diferentes lenguajes y herramientas de programación.*
3. *Aplicar metodologías de desarrollo de software, para la producción de códigos que cumplan con los estándares de la industria.*
4. *Implementar pruebas y evaluaciones de software, para garantizar el correcto despliegue y mantenimiento del producto o artefacto de software.*
5. *Administrar proyectos de software, para avalar la calidad y viabilidad del desarrollo en estudio.*
6. *Conformar equipos de trabajo dinámicos, aplicando metodologías ágiles.*

COMPETENCIAS ESPECÍFICAS Y CRITERIOS DE REALIZACIÓN

Competencia Profesional 1: *Analizar requerimientos y diseñar soluciones de software, comprendiendo las especificaciones funcionales, no funcionales e infraestructura tecnológica.*

1.1

Competencias Específicas	Criterios de realización
a) Analizar requerimientos e información	• Se solicitan las especificaciones funcionales, no funcionales e infraestructura tecnológica, estableciendo las características de una solución software que satisfaga la solicitud de los requerimientos. • Se considera la diversidad de contextos y/o plataformas informáticas en las que se integran dichos requerimientos.

Competencia Profesional 2: *Desarrollar software utilizando diferentes lenguajes y herramientas de programación.*

2.1

Competencias específicas	Criterios de realización
a) Diseñar la arquitectura del producto software a desarrollar	• Se consideran los distintos contextos e infraestructuras existentes, para diferentes tipos de aplicaciones según las tecnologías usadas. • Se genera la documentación y diagramas adecuados para los tipos de arquitectura. • Se demuestra capacidad para viabilizar un proyecto de desarrollo de un producto software.

2.2

Competencias específicas	Criterios de realización
a) Desarrollar artefactos de software compartidos, reutilizables e integrables	• Se implementa el modelo diseñado, aplicando patrones de diseño o reutilizando arquitecturas existentes, pudiendo integrar artefactos de terceros o diseñando los propios. • Se emplean adecuadamente herramientas de desarrollo de software individual y herramientas de desarrollo de software en comunidad • Se utilizan herramientas de gestión y seguimiento de las versiones de los artefactos software que componen el producto software construido o desarrollado.

Competencia Profesional 3: *Aplicar metodologías de desarrollo de software, para la producción de códigos que cumplan con los estándares de la industria.*

3.1.

Competencias específicas	Criterios de realización
a) Generar y producir código de programación	• Se produce código que se ejecute y se pueda desplegar en distintos ambientes tecnológicos con diferentes sistemas informáticos. • Se utilizan generadores de código automáticos y generadores de algoritmos automáticos. • Se verifica, valida, revisa e inspecciona el código generado o escrito por máquinas o personas.

3.2.

Competencias específicas	Criterios de realización
a) Resolver problemas técnicos específicos	• Se resuelven problemas poniendo en juego la capacidad de identificar y resolver de manera efectiva, con habilidades técnicas y competencias en acción de forma eficiente. • Se conocen las herramientas y tecnologías utilizadas para expresar las soluciones encontradas. • Se demuestran capacidades de trabajo con precisión y atención, para garantizar que el software se construya de manera correcta y con calidad, eliminando inconsistencias que terminen en fallas latentes y manifestación de errores posteriores. • Se demuestran cualidades de un Tech lead referente que tiene un alto nivel de conocimiento técnico y liderazgo en el equipo de desarrollo. • Se guía a los miembros del equipo en la toma de decisiones técnicas y en la implementación de soluciones eficientes y útiles.

Competencia Profesional 4: *Implementar pruebas y evaluaciones de software, para garantizar el correcto despliegue y mantenimiento del producto o artefacto de software.*

4.1.

Competencias específicas	Criterios de realización
a) Gestionar proyectos de software	• Se asiste con el correcto proceso del desarrollo y despliegue y mantenimiento del producto software resultante. • Se gestionan los artefactos de software propios o ajenos, teniendo en cuenta las necesidades de cobertura de las diferentes pruebas e integración continua, diseño y ejecución, registrando el versionado. • Se colabora en estrecha relación con los miembros del equipo para realizar acciones correctivas sobre el código generado o cualquier artefacto de software producido o usado. • Se calculan los tiempos y costos de un proyecto software.

4.2.

Competencias específicas	Criterios de realización
a) Migrar a contextos y pilas tecnológicos diferentes	• Se demuestra un perfil versátil y amplio en conocimientos técnicos específicos en FullStack, DevOps, Virtualización y Nube. • Se manifiesta adaptación rápida a los cambios y requerimientos del mercado laboral y tecnológico, con habilidades de desempeño en entornos virtuales, presenciales y mixtos. • Se fomenta el teletrabajo de forma remota y se colabora con equipos distribuidos en diferentes partes del mundo.

Competencia Profesional 5: *Administrar proyectos de software para garantizar la calidad y viabilidad del desarrollo en estudio.*

5.1.

Competencias específicas	Criterios de realización
a) Documentar las actividades productos del desarrollo	• Se gestiona el proyecto y viabilidad del desarrollo de los procesos involucrados en la construcción de un producto software, empleando herramientas, normas y estándares que indica la industria. • Se utilizan modelos, métodos, técnicas, procedimientos, actividades y herramientas adecuadas en la gestión de proyectos de software. • Se oficia la calidad, versiones y repositorios de artefactos software.

5.2.

Competencias específicas	Criterios de realización
a) Realizar la autogestión en el contexto del desarrollo de software	• Se organiza y dirige su propio trabajo de manera efectiva sin una supervisión constante. • Se establecen formas de trabajo claras, administrando los tiempos de manera eficiente, tomando decisiones y resolviendo problemas con criterios fundamentados e independientes. • Se utilizan programas de gestión de proyectos. • Se autogestionan entornos ágiles, dinámicos y cambiantes de trabajo. • Se demuestra la adaptación rápida a los cambios y a la toma de decisiones de manera autónoma.

Competencia Profesional 6: *Conformar equipos de trabajo dinámicos, aplicando metodologías ágiles.*

6.1

Competencias específicas	Criterios de realización
a) Establecer relaciones interpersonales y comunicación efectiva	• Se potencia el trabajo en equipo y colaborativo en los múltiples ámbitos de desempeño que plantea el mundo laboral, adaptándose a las situaciones y demandas del contexto cambiante. • Se manifiestan capacidades de comunicación efectiva con los miembros del equipo, usando entornos colaborativos presenciales, virtuales, a distancia y mixtos, para trabajar con otros miembros del equipo. • Se establecen objetivos comunes, se comparten conocimientos, habilidades y se demuestran habilidades de relación con los stake holders del proyecto, que implican procesos sinérgicos y creativos.

III.COMPONENTES CURRICULARES

1. Organización curricular por campos de formación

Campos	Nombre	Régimen de Cursado	Horas cátedra anuales	Horas reloj anuales
GENERAL	Inglés técnico I	Anual	120	80
	Alfabetización académica	Cuatrimstral	60	40
	Legislación Informática y Ética profesional	Cuatrimstral	45	30
Subtotal			225	150
% del Campo				8,1
DE FUNDAMENTO	Matemática Aplicada	Anual	90	60
	Lógica Computacional	Anual	90	60
	Arquitectura de Dispositivos	Cuatrimstral	45	30
	Seminario Nuevas Tecnologías	Cuatrimstral	45	30
	Matemática Discreta	Cuatrimstral	60	40
	Análisis Matemático	Cuatrimstral	60	40
	Inglés técnico II	Anual	120	80
	Inglés técnico III	Anual	120	80
	Estadística y Probabilidades para el Desarrollo de Software	Cuatrimstral	60	40
Subtotal			690	460
% del Campo				24,7
ESPECÍFICO	Programación I	Anual	180	120
	Sistemas Operativos I	Cuatrimstral	45	30
	Base de Datos I	Anual	90	60
	Programación II	Anual	180	120
	Comunicación y Redes	Anual	90	60
	Modelado de Software	Anual	120	80
	Sistemas Operativos II	Anual	90	60
	Base de Datos II	Anual	90	60
	Programación III	Anual	120	80
	Arquitectura y Diseño de Interfaces: UI y UX	Anual	120	80
	Auditoría y Calidad de Sistemas	Cuatrimstral	45	30
	Ciberseguridad	Cuatrimstral	45	30
	Gestión de Proyectos de Software	Cuatrimstral	45	30
	Metodologías Ágiles	Cuatrimstral	45	30
	Base de Datos III	Anual	90	60
Subtotal			1395	930
% del Campo				50,0
	Práctica Profesionalizante I	Anual	120	80
	Práctica Profesionalizante II	Anual	180	120

	Práctica Profesionalizante III	Anual	180	120
Subtotal			480	320
% del Campo				17,2
Total de horas			2790	1860

2. Distribución de espacios curriculares por año.

PRIMER AÑO

1° Cuatrimestre					2° Cuatrimestre				
Unidad Curricular		Formato	HC	HCT	Unidad Curricular		Formato	HC	HCT
Cód.	Denominación				Cód.	Denominación			
1	Programación I	TA	6		1	Programación I	TA	6	180
2	Matemática Aplicada	A	3		2	Matemática Aplicada	A	3	90
3	Lógica Computacional	M	3		3	Lógica Computacional	M	3	90
4	Inglés técnico I	T	4		4	Inglés técnico I	TA	4	120
5	Alfabetización académica	T	4	60	6	Seminario Nuevas Tecnologías	S	3	45
7	Arquitectura de Dispositivos	M	3	45	8	Sistemas Operativos I	M	3	45
9	Base de Datos I	T	3		9	Base de Datos I	T	3	90
10	Práctica Profesionalizante I	pp	4		10	Práctica Profesionalizante I	PP	4	120
Total de horas Cátedra 1° Cuatrimestre			30		Total de horas Cátedra 2° Cuatrimestre			29	
Total de horas cátedra de Primer Año									885
Total de horas reloj de Primer Año									590

SEGUNDO AÑO

1° Cuatrimestre					2° Cuatrimestre				
Unidad Curricular		Formato	HC	HCT	Unidad Curricular		Formato	HC	HCT
Cód.	Denominación				Cód.	Denominación			
11	Programación II	T	6		11	Programación II	T	6	180
12	Matemática Discreta	A	4	60	13	Análisis Matemático	A	4	60
14	Comunicación y Redes	M	3		14	Comunicación y Redes	M	3	90
15	Inglés técnico II	T	3		15	Inglés técnico II	T	3	120
16	Modelado de Software	M	4		16	Modelado de Software	M	4	120
17	Sistemas Operativos II	M	3		17	Sistemas Operativos II	M	3	90
18	Base de Datos II	T	3		18	Base de Datos II	T	3	90
19	Práctica Profesionalizante II	PP	6		19	Práctica Profesionalizante II	PP	6	180
Total de horas Cátedra 1° Cuatrimestre			32		Total de horas Cátedra 2° Cuatrimestre			32	
Total de horas cátedra de Segundo año									990
Total de horas reloj de Segundo año									660

TERCER AÑO

1° Cuatrimestre					2° Cuatrimestre				
Unidad Curricular		Formato	HC	HCT	Unidad Curricular		Formato	HC	HCT
Cód.	Denominación				Cód.	Denominación			
20	Programación III	T	4		20	Programación III	T	4	120
21	Arquitectura y Diseño de Interfaces: UI y UX	M	4		21	Arquitectura y Diseño de Interfaces: UI y UX	M	4	120
22	Auditoría y Calidad de Sistemas	M	3	45	23	Ciberseguridad	M	3	45
24	Inglés técnico III	T	3		24	Inglés técnico III	T	3	120
25	Legislación Informática y Ética profesional	M	3	45	26	Estadística y Probabilidades para el Desarrollo de Software	T	4	60
27	Gestión de Proyectos de Software	T	3		28	Metodologías Ágiles	T	3	90
29	Base de Datos III	T	3		29	Base de Datos III	T	3	90
30	Práctica Profesionalizante III	PP	6		30	Práctica Profesionalizante III	PP	6	180
Total de horas Cátedra 1° Cuatrimestre			29		Total de horas Cátedra 2° Cuatrimestre			30	
Total de horas cátedra de tercer año									915
Total de horas reloj de tercer año									610
Total de horas cátedra de la carrera									2790
Total de horas reloj de la carrera									1860

3. Trayectorias Formativas para Certificaciones Intermedias Formación Profesional de Nivel 3 (FP3)

Denominación: Programador Junior / Desarrollador Full Stack Junior

3.1. Perfil profesional de la Certificación

- ✓ Interpretación del problema y convalidación de su interpretación a fin de identificar aspectos faltantes.
- ✓ Las especificaciones de diseño o requisitos de las asignaciones a programar, comprendiendo en su contexto inmediato cuál es el problema para resolver.
- ✓ Definición del alcance y Desarrollo de algoritmos que dan soluciones a los problemas asignados o derivados de los mismos.

- **Características y Habilidades Comunes en un Programador Junior**

Un Programador Junior debe tener conocimientos técnicos básicos, habilidades blandas en desarrollo y una disposición para aprender y crecer en el campo de la programación. Esto se logrará al finalizar y aprobar todas las materias de la Tecnicatura Superior en Desarrollo de Software

CONOCIMIENTOS TÉCNICOS

1. **Fundamentos de Programación:** Comprensión básica de conceptos como variables, tipos de datos, estructuras de control, funciones y métodos.
2. **Lenguajes de Programación:** Conocimiento básico de al menos un lenguaje de programación, como Java, Python, C#, JavaScript, etc.
3. **Algoritmos y Estructuras de Datos:** Comprensión de algoritmos básicos y estructuras de datos como listas, pilas, colas, árboles y grafos.
4. **Bases de Datos:** Conocimientos básicos de bases de datos relacionales y SQL.
5. **Desarrollo Web:** Familiaridad con tecnologías web básicas como HTML, CSS y JavaScript.
6. **Control de versiones:** Uso básico de sistemas de control de versiones como Git.

HABILIDADES BLANDAS

1. **Aprendizaje Continuo:** Disposición para aprender y adaptarse a nuevas tecnologías y herramientas.
2. **Resolución de Problemas:** Capacidad para identificar problemas y encontrar soluciones creativas.
3. **Comunicación:** Mantener una comunicación efectiva para garantizar que la información fluya de manera armoniosa dentro de la empresa y entre los distintos estamentos jerárquicos. Para ello, se necesita poner en acción la escucha activa, claridad en el mensaje, uso de lenguaje simple y directo, producción mensaje según las necesidades, evitando ambigüedades para buscar entender el punto de vista del interlocutor.
4. **Trabajo en Equipo:** Actitud para colaborar y gestionar en equipo.
5. **Atención al detalle:** Habilidad para prestar atención a los detalles y escribir código limpio y legible.
6. **Pensamiento Lógico:** Implica la capacidad de analizar información, identificar patrones, establecer relaciones causa-efecto y argumentar de forma válida con coherencia, la inferencia y la consistencia

EXPERIENCIA Y PROYECTOS

1. **Proyectos Personales:** Pueden tener proyectos personales o de estudio que demuestren su capacidad para aplicar los conocimientos adquiridos.
2. **Prácticas o Internships:** Experiencia en prácticas o internships en empresas de tecnología.
3. **Participación en Comunidades:** Participación activa en comunidades de desarrollo, foros, grupos de estudio, etc.

3.2. Componentes curriculares de la certificación:

Designación de certificación Intermedia	Espacios Curriculares acreditados	HC
Programador Junior / Desarrollador Full Stack Junior	Programación I	180
	Matemática Aplicada	90
	Lógica Computacional	90
	Inglés técnico I	120
	Alfabetización académica	60
	Arquitectura de Dispositivos	45
	Sistemas Operativos I	45
	Bases de Datos I	120
	Seminario Nuevas Tecnologías	45
	Práctica Profesionalizante I	120
	Comunicaciones y Redes	90
	Programación II	120
	Inglés técnico II	120
	Modelado de Software	120
	Base de Datos II	90
	Sistemas Operativos II	90
	Práctica Profesionalizante II	180
Carga Horaria Total		1725

3.3. Requisitos para la certificación de la trayectoria

Se otorgará la certificación correspondiente a los estudiantes que acrediten los espacios curriculares de la trayectoria y que a través de un trabajo integrador final específico al cierre del proceso demuestren haber logrado las competencias definidas por perfil profesional.

El trabajo integrador específico deberá surgir de las prácticas profesionalizantes I y II y deberá ser inherente a la certificación otorgada.

En la dicha instancia de acreditación deberán constituirse un tribunal evaluador integrado por el coordinador de la carrera y un profesor por cada campo de formación (general, fundamento, específica y práctica profesionalizante).

4. DESCRIPTORES POR ESPACIO CURRICULAR

PRIMER AÑO

1 - PROGRAMACIÓN I

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.

Concepto de lenguaje de alto nivel y la necesidad de traducción, comparación entre compiladores e intérpretes, aspectos de la traducción dependientes y no dependientes de la máquina. Máquinas virtuales, concepto, jerarquía de máquinas virtuales, lenguajes intermedios, asuntos de seguridad que surgen al ejecutar código en una máquina diferente. Representación de datos numéricos, rango, precisión y errores de redondeo. Arreglos. Representación de datos de caracteres, listas y su procesamiento.

Programación modular: Concepto. Aplicación: estructura de un programa utilizando procedimientos y funciones.

Reglas para escribir algoritmos eficientes. Elaboración de "algoritmos-tipo" o estándar a partir de métodos lógicos matemáticos, por ejemplo: uso de funciones matemáticas recursivas, funciones recursivas simples, búsqueda sucesiva y binaria y de ordenamiento. Algoritmos de camino mínimo. Elementos de complejidad de algoritmos. Pruebas de escritorio para validar algoritmos. Verificación unitaria de unidades de código, concepto de cubrimiento, organización, ejecución y documentación de la prueba.

Desarrollo de Programas Ambientes de programación. Introducción al uso de librerías y APIs (interfaz de programación de aplicaciones). Lenguaje de programación: Estructura sintáctica de un programa en el lenguaje de aplicación. Reglas sintácticas del lenguaje. Sintaxis de procedimientos y funciones. Reglas del lenguaje.

Programación Web Básica Fundamentos de la programación web: HTML, CSS y JavaScript. Creación y estructuración de páginas web utilizando HTML. Estilos y diseño de páginas web con CSS. Interactividad en páginas web con JavaScript. Desarrollo de formularios y validación de datos. Introducción a las bibliotecas y frameworks de JavaScript. Buenas prácticas en el desarrollo web. Pruebas y depuración de aplicaciones web.

2 – MATEMÁTICA APLICADA

Matrices y Determinantes. Conjuntos numéricos. Matrices. Concepto de matriz. Dimensión de una matriz. Tipos de matrices: matriz fila, matriz columna, matriz cuadrada, matriz rectangular, matriz diagonal, matriz simétrica. Igualdad de matrices. Operaciones con matrices. Determinantes: La función determinante, matriz inversa, rango. Espacio Vectorial. Vectores de n componentes. Generalización. Operaciones internas y externas, normas, proyecciones, dependencia lineal, base y dimensión. Adición y sustracción de vectores. Producto de un vector por un escalar. Producto escalar.

Sistema de ecuaciones. Clasificación. Teorema de Roché Frobenius, resolución. Expresión matricial de un sistema ecuaciones lineales con N incógnitas. Sistemas equivalentes. Resolución de sistemas de ecuaciones. Compatibilidad de los sistemas. Sistemas homogéneos. Operaciones internas y externas, norma, proyecciones, dependencia lineal, base y dimensión. Aplicaciones de los espacios vectoriales. Transformaciones lineales: teorema fundamental, matriz asociada,

autovalores y autovectores, diagonalización.

3 – LÓGICA COMPUTACIONAL

Lógica proposicional. Elementos de lógica. Lógica proposicional, conectivos lógicos. Formas normales: conjuntiva y disyuntiva. Validez. Adquisición del conocimiento, forma del conocimiento, uso del conocimiento, límites del conocimiento. Intratabilidad e inexpresabilidad. Enunciados y conectivas. Funciones de verdad y tablas de verdad. Argumentación y validez. Lógica de Enunciados. Reglas de manipulación y sustitución. Formas normales. Conjuntos adecuados de conectivas.

Lógica de predicados. Lógica de predicados, cuantificadores: Universal y existencial. Limitaciones de la lógica de predicados- Lenguajes de primer orden. Interpretaciones Satisfacción y verdad. El sistema formal. Corrección y completitud. Modelos de sistemas de primer orden.

Lógica digital. Introducción a la Lógica digital, Álgebra de Boole, Compuertas lógicas: NAND (No Y), NOR (No O), OR exclusiva (O exclusiva). Multiplexores, decodificadores, biestables, memorias, microcontroladores, microprocesadores. Funciones. Método de Karnaugh

4 – INGLÉS TÉCNICO I

Los siguientes descriptores deberán abordarse desde las cuatro macro-habilidades (comprensión lectora y auditiva y producción escrita y oral) y orientarse al campo de desempeño del futuro profesional, desarrollando las competencias básicas fijadas por los estándares del Marco Común Europeo de Referencia para la competencia lingüística de nivel A1: al finalizar el cursado, el estudiante será capaz de comprender y utilizar expresiones cotidianas de uso muy frecuente así como frases sencillas destinadas a satisfacer necesidades de tipo inmediato. Podrá presentarse a sí mismo y a otros, pedir y dar información personal básica sobre su domicilio, sus pertenencias y las personas que conoce. Podrá relacionarse de forma elemental siempre que su interlocutor hable despacio y con claridad y esté dispuesto a cooperar.

Contenidos mínimos:

El estudiante deberá desarrollar habilidades de producción (habla y escritura) aplicadas a situaciones específicas de la carrera y el año que transita.

Preguntas abiertas. Preguntas cerradas. Sustantivos y adjetivos posesivos. Verbo be. Preposiciones de tiempo y lugar. Preferencias con would like. Presente simple. There is y there are. Sustantivos contables e incontables. Artículos definido e indefinido. Presente continuo. Verbos modales. Pasado simple. Adjetivos comparativos y superlativos. Futuro con be going to.

5 – ALFABETIZACIÓN ACADÉMICA

Competencia comunicativa. La comunicación: definición, elementos de la situación comunicativa. Funciones del lenguaje. El contexto: adecuación del texto al contexto. Registro. Objetividad y subjetividad en el mensaje. Tipos de comunicación. Concepto de texto. Concepto de discurso.

Propiedades de los textos: adecuación, coherencia, cohesión y normativa. Tipologías textuales. Texto explicativo-expositivo. Texto argumentativo. Texto instructivo. Correcta redacción de prompts de IA, preguntas y respuestas en bots, y lenguajes de tipo low-code más textuales.

La comprensión lectora: estrategias y fases. El resumen/síntesis. Producción de Textos: planificación, textualización y revisión. Informe. Textos instrumentales. La intencionalidad comunicativa: persuasión e información. Reconocimiento de ideas nucleares y periféricas. Elaboración de esquemas: jerárquicos – cronológicos – comparativos. Indicadores de autoevaluación.

Introducción al procesamiento de lenguaje natural (PLN), relación con la inteligencia artificial. Importancia de la claridad y precisión en la redacción de prompts para obtener respuestas adecuadas. Estrategias para adaptar el tono y estilo según el contexto, junto con técnicas para identificar y corregir problemas en la interpretación de prompts por parte de la IA. Evaluación de prompts y consideraciones éticas para evitar sesgos en la interacción con herramientas de IA.

Capacidad de aprender a aprender. Learn agility. Definiciones. La necesidad del aprendizaje continuo en los tiempos del Agilísimo y la digitalización. Aprender a reaprender. Contexto VUCA. Contexto BUNI. Aprendizaje y trabajo. Upskilling. Reskilling. Estilos de aprendizaje. Actitudes del aprendiz. Aprender a ser aprendiz. Mejorar la capacidad de aprender. Inteligencia interpersonal e intrapersonal. Inteligencia emocional. Herramientas para desarrollar la Learnability individual. Rasgos clave. Activar el proceso de Aprendizaje Ágil. Capacidad de aprender sobre el error. Experimentar. Gestión del feedback: pedido y escucha. Gestión de la motivación intrínseca.

6 – SEMINARIO NUEVAS TECNOLOGÍAS

Desarrollo de Software: Introducción al desarrollo de software y su impacto en la sociedad moderna y en las organizaciones. Las metodologías de desarrollo de software utilizadas en la actualidad: lenguajes, entornos, frameworks, modelos de desarrollo. El desarrollo de aplicaciones standalone, cliente-servidor, y similares. La programación de dispositivos móviles, de componentes IoT y otros. Elementos de integración de componentes mediante APIs. Las tendencias en el desarrollo de software.

Inteligencia Artificial. Introducción a la ciencia de datos e inteligencia artificial. Relevancia de la IA en el desarrollo de software, en la telemática y en la ciencia de datos. Herramientas populares: GitHub Copilot, TabNine, ChatGPT y otras generativas. Las aplicaciones más utilizadas de la IA en otras profesiones y actividades: en medicina, ingeniería, finanzas, etc. Tendencias de la IA y la Ciencia de Datos.

Redes de Computadoras y Ciber seguridad: Introducción a las comunicaciones de datos y su influencia en las sociedades modernas. La seguridad informática, los riesgos actuales y su impacto en las organizaciones y en la sociedad. Las tecnologías de comunicaciones de datos y su estado actual: dispositivos, medios físicos, estándares de velocidad y transporte. Tendencias en las comunicaciones de datos y en la seguridad. Delitos informáticos, impacto y su investigación.

7 – ARQUITECTURA DE DISPOSITIVOS

Sistemas numéricos de distintas bases, operaciones básicas, resta por complemento, circuitos lógicos y digitales básicos, códigos y representaciones. Tecnología: memorias, almacenamientos auxiliares, dispositivos de entrada y salida. Unidad Central de Procesamiento (CPU). Estructura y funcionamiento de la CPU. Memoria. Tipos de memoria: RAM, ROM, Cache.

Jerarquía de memoria. Funcionamiento de la memoria virtual. Sistemas de Almacenamiento. Discos duros (HDD) y unidades de estado sólido (SSD). Sistemas de archivos y gestión de almacenamiento. Buses de Datos y Direcciones Unidades de Entrada y Salida (E/S). Bloque Arquitectura Interna de la CPU y Sistemas de Memoria. Arquitectura Interna de la CPU. Registro y ALU (Unidad Aritmética Lógica). Tipos de registros. Funcionamiento de la ALU. Unidad de Control. Sistemas de memoria. Memoria caché. Memoria principal.

Simulación de Arquitectura y Montaje. Introducción a BLUE y Simulación de Arquitectura. Instalación y configuración de BLUE. Interfaz de usuario de BLUE. Simulación de arquitectura de computadoras con BLUE. Prácticas con BLUE. Montaje y desmontaje de un dispositivo. Identificación y descripción de componentes físicos.

8 – SISTEMAS OPERATIVOS I

Fundamentos. Introducción a los sistemas operativos. Historia y evolución de los sistemas operativos. Funciones y componentes de un sistema operativo. Tipos de sistemas operativos. Arquitectura de sistemas operativos. Gestión de procesos: Estados de un proceso. Planificación de procesos. Hilos y multitarea. Gestión de memoria: Jerarquía de memoria. Memoria principal y secundaria. Memoria virtual. Técnicas de asignación y paginación. Sistemas de archivos: Estructura y organización de sistemas de archivos. Tipos de sistemas de archivos. Operaciones sobre sistemas de archivos. Gestión de dispositivos: Controladores de dispositivos. Gestión de entradas y salidas (E/S).

Administración. Instalación y configuración de sistemas operativos. Administración de usuarios y permisos. Gestión de recursos: Administración de memoria. Administración de procesos. Administración de dispositivos. Seguridad en sistemas operativos: Políticas de seguridad. Mecanismos de autenticación y autorización. Protección contra malware. Copias de seguridad y recuperación de datos. Monitoreo y optimización del rendimiento del sistema.

Programación de Sistemas Operativos. Programación en el espacio del usuario y del kernel. Llamadas al sistema (system calls). Programación de scripts de shell: Shell scripting en Unix/Linux. Comandos y utilidades básicas. Automatización de tareas. Desarrollo de controladores de dispositivos. Programación concurrente: Hilos y sincronización. Exclusión mutua y comunicación entre procesos. Manejo de señales y excepciones. Desarrollo de sistemas de archivos. Prácticas de depuración y manejo de errores.

9 – BASE DE DATOS I

Organización de Datos. Modelos conceptuales (E/R, UML), modelo orientado a objetos, modelo relacional, modelos semiestructurados (XML). Componentes y funciones de un sistema de base de datos. Definición de datos, álgebra relacional. Estructuras de almacenamiento. Modelo Entidad-Relación: Entidad. Relaciones entre entidades. Atributo de las entidades.

Diseño de base de datos. Diseño lógico y diseño físico. 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.

Fundamentos de Administración y Gestión de Base de Datos. Sistema Gestor de base de datos. Actores y roles del entorno. Recuperación de la información. Gestión de bases de datos. Accesos, permisos y roles. Creación de vistas e índices. Lenguaje SQL/ MySQL y otros. Operaciones: consultas, alta, baja y modificación de registros. Procedimientos almacenados. Disparadores. Usuarios. Transacciones.

10 – PRÁCTICA PROFESIONALIZANTE I

Las Prácticas Profesionalizantes plantean estrategias y actividades formativas que tienen como propósito que los estudiantes consoliden e integren las capacidades o saberes que se corresponden con el perfil profesional en el que se están formando. El primer espacio curricular de práctica profesionalizante pretende iniciar a los estudiantes en el proceso de construcción del rol profesional del Técnico Superior en Desarrollo de Software (TSDS), que se irá proyectando y profundizando durante la carrera. Para cumplir este objetivo, las primeras prácticas estarán referidas a: - Conocer los diferentes contextos de trabajo.

- Identificar diferentes procesos de trabajo, sus características, variables puestas en juego, en ambientes reales de trabajo y/o simulados.
- Reconocer los diferentes actores que conforman el campo profesional.

Todas estas actividades, que se podrán realizar de acuerdo a las posibilidades de articulación institucional con el sector tanto dentro de la institución como fuera de ella, permitirán que los estudiantes tengan una visión más completa e integral sobre el campo profesional, sus características, la diversidad de contextos de intervención, las diferentes relaciones que se ponen en juego, las tensiones y los conflictos que pueden aparecer. También proponen abordar en forma práctica los procesos vinculados a la interacción entre los diversos actores que conforman un equipo de trabajo, identificando los roles y responsabilidades de cada uno y su relación con la tarea profesional del TSDS, profundizando y reflexionando sobre su rol con responsabilidad legal y social. Este espacio contempla 120 horas que podrán incluir, por un lado, la aproximación del estudiante a experiencias directas y visitas a diversos ámbitos de trabajo con el fin de realizar observaciones, entrevistas, encuestas (métodos de recolección de información) que le permitan conocer las características de los contextos laborales, las vinculaciones con otros actores del sector y/o equipos

de trabajo, profundizando y reflexionando sobre las funciones específicas del TSDS. Por otra parte, este espacio también debe comprender horas de trabajo áulico bajo la coordinación del docente, quien podrá implementar diferentes estrategias que le permitan simular todas las etapas de trabajo de creación de software. En este espacio áulico, los estudiantes deberán poner en común e intercambiar con sus compañeros las particularidades de cada simulación realizada, de cada ámbito relevado, el impacto de los proyectos y de los roles asumidos con el fin de promover el debate y el proceso de aprendizaje colectivo. Para orientar la evaluación, se proponen algunos indicadores que pueden ser utilizados como evidencias, a partir de las cuales se podrá inferir si los estudiantes han alcanzado los objetivos propuestos:

- Comprende el rol profesional del TSDS y sus incumbencias.
- Identifica las características/etapas del proceso de creación de software y elabora los informes correspondientes.
- Reconoce las funciones de los diferentes actores intervinientes en dicho proceso.

SEGUNDO AÑO

11 - PROGRAMACIÓN II

Introducción al paradigma de objetos. Comparación entre paradigmas. Clases y objetos. Subclases. Atributos. Métodos. Modularidad. Modificadores de visibilidad. Encapsulamiento Sobrecarga de métodos. Concepto de acoplamiento. Herencia. Sobreescritura. Clases abstractas... Agregación/composición: por valor y por referencia. Asociación: multiplicidad. Atributos y métodos estáticos. Modelo de tres capas. Administración de excepciones. El modelo computacional de objetos. Tiempo de vida de los objetos. Diseño UML. Diagrama de clases. Relaciones entre clases: herencia, asociación, composición y agregación. Diagrama de Casos de Uso. Diagrama de Secuencia. Construcción de un diseño.

Desarrollo de Sistemas. Lenguaje basado en clases. Lenguajes Visuales vs. Lenguajes Orientados a Objetos. (Java. Python. Aplicaciones Mobile -Kotlin-) Estructura de una clase. Variables de instancia. Métodos de acceso y de modificación. Constructores. Constructores y métodos sobrecargados. Métodos accesorios o auxiliares. Documentación de clases y de métodos. Librería de clases del lenguaje. Clases de fines específicos. Agrupamiento de objetos. Manejo de errores. Clase de prueba. Constantes y variables de clase. Interfaz e implementación de una clase. Noción de evento. Interfaz de usuario.

12 – MATEMÁTICA DISCRETA

SISTEMAS NUMÉRICOS Y MÉTODOS DE CONTEO. Introducción. Tipos de sistemas numéricos. Conversiones. Operaciones básicas.

INTRODUCCIÓN A LA TEORÍA DE NÚMEROS. divisibilidad. Números primos. Máximo común divisor y mínimo común múltiplo. Algoritmo de la división en los enteros. Algoritmo de Euclides.

Primos relativos. Aritmética modular. Relaciones de Congruencia.

ARITMÉTICA MODULAR. Relaciones de Congruencia. Aritmética en (suma, resta, multiplicación, división, elementos inversibles y potencia). Ecuaciones en. Teorema chino de los restos. El pequeño Teorema de Fermat. Expresión matemática de algoritmos de cifrado simétrico, aritmética modular aplicada y curva elíptica. Conformación de claves RSA.

13 – ANÁLISIS MATEMÁTICO

Funciones. Tipos Relaciones; Tipos Conjuntos; Números Reales. Intervalos y entornos. Conjuntos: clasificación. Entorno y entorno reducido. Punto de acumulación. Funciones de variable real. Clasificación. Álgebra de funciones continuas. Composición de funciones. Concepto de límite y funciones. Límites finitos e infinitos. Continuidad. Cálculo de derivadas. Derivada en un punto. Recta tangente. Función derivada. Determinación de constantes. Aplicación. Cálculo de integrales. Integral definida. Áreas y volúmenes de revolución.

14 – COMUNICACIÓN Y REDES

Fundamentos de las redes de comunicación, comprendiendo su estructura y arquitectura, los protocolos utilizados, el modelo ISO/OSI y los diferentes tipos de redes existentes. Posteriormente, se profundizará en el diseño y administración de redes, abarcando aspectos como la topología de red, la seguridad informática y la gestión de recursos. Además, se explorarán las tecnologías y herramientas utilizadas en el campo de las comunicaciones, incluyendo los sistemas de transmisión de datos, los protocolos de enrutamiento y las técnicas de multiplexación. Finalmente, se analizarán las tendencias actuales en el ámbito de las redes, como el Internet de las cosas (IoT) y las redes de área amplia (WAN), brindando a los estudiantes una visión integral de la comunicación y las redes en el mundo del desarrollo de software. Modelo OSI del Protocolo Tcp/ip y sus variantes - sub protocolos en cada capa del mismo. Ejemplo Http, Icmp, Ftp, etc

Características de la programación concurrente: Ventajas. Proceso. Hilo de ejecución. Hilos y procesos múltiples. Sincronización de procesos. Exclusión mutua de procesos. Comunicación entre procesos. Prioridades de los procesos. Sistemas distribuidos: concepto, características, Modelo Cliente-Servidor, Aplicación Cliente, Aplicación Servidor. Sockets: Concepto y características de los sockets. Tipos de sockets. Sockets de flujo o TCP. Operaciones sobre el TCP. Cliente y el TCP servidor. Intercambio de flujo de datos. Sockets de datagramas o UDP. Operaciones sobre el UDP cliente y el UDP servidor. Envío y recepción de datagramas. Bloque Seguridad en desarrollo en redes. Seguridad en aplicaciones cliente servidor: Alcance, puntos de vulnerabilidad. Técnicas de seguridad para aplicaciones en cliente y aplicaciones en servidor. Herramientas de detección y prevención de vulnerabilidades en las aplicaciones. Control de acceso a las aplicaciones y al servidor. Detección de accesos- Monitoreo de servidor: Auditorías. Control de versiones.

Desarrollo de aplicaciones. Flujos de datos. Manejo de archivos en medios físicos. Clases para el manejo de archivos. Clases para el manejo de flujos a través de la red. Buffer: concepto y uso.

Captura de errores en los sockets. Clases para la programación multihilo. Clases para implementar aplicaciones sobre redes: utilizando TCP, utilizando UDP, para gestionar una dirección IP, Servidor local: concepto y uso, Servidor remoto: concepto y uso., Servicios web

15 – INGLÉS TÉCNICO II

Los siguientes descriptores deberán abordarse desde las cuatro macro-habilidades (comprensión lectora y auditiva y producción escrita y oral) y orientarse al campo de desempeño del futuro profesional, desarrollando las competencias básicas fijadas por los estándares del Marco Común Europeo de Referencia para la competencia lingüística de nivel A2: al finalizar el cursado, el estudiante será capaz de comprender frases y expresiones de uso frecuente relacionadas con áreas de experiencia que le son especialmente relevantes (información básica sobre sí mismo y su familia, compras, lugares de interés, ocupaciones, etc). Sabrá comunicarse a la hora de llevar a cabo tareas simples y cotidianas que no requieran más que intercambios sencillos y directos de información sobre cuestiones que le son conocidas o habituales. Podrá describir en términos sencillos aspectos de su pasado y su entorno así como cuestiones relacionadas con sus necesidades inmediatas.

Contenidos mínimos:

El estudiante deberá desarrollar habilidades de producción (habla y escritura) aplicadas a situaciones específicas de la carrera y el año que transita.

Futuro con will. Pasado continuo. Presente perfecto. Presente perfecto continuo. Verbos modales. Condicional 0, 1 y 2. Voz pasiva. Gerundios e infinitivos.

16 – MODELADO DE SOFTWARE

Modelo de Objetos, modelo dinámico, modelo funcional. Análisis de requerimientos de software, el proceso, partes interesadas. Requerimientos funcionales, prioridades y criterios de realización. Análisis orientado a objetos y UML. Diagramas de clase. Escenarios, historias y casos de uso. Representación del comportamiento: diagramas de secuencia, máquinas de estado, diagramas de actividad. Representación del comportamiento: diagramas de secuencia, máquinas de estado, diagramas de actividad. Pre y Post condiciones.

Metodología de diseño Orientado a Objetos. Principios generales de diseño: descomposición, desacoplamiento, cohesión, reusabilidad, portabilidad, testabilidad, flexibilidad, escalabilidad. Patrones de diseño. Arquitecturas de software: concepto de vistas. 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.

Relación entre patrones de diseño, arquitectónicos y estilos arquitectónicos. Principios de diseño S.O.L.I.D: el Principio de responsabilidad única (SRP), el Principio de abierto-cerrado (OCP), el Principio de sustitución de Liskov (LSP), el Principio de segregación de la interfaz (ISP), el Principio de inversión de la dependencia (DIP). Principios de componentes: cohesión de los componentes, acoplamiento de componentes. Atributos de calidad. Importancia de los atributos de calidad.

Clasificación de los atributos de calidad. Atributos de calidad observables y no observables. Patrones de diseño de software, criterios para la selección y aplicación de los patrones, rol de los patrones de diseño en la solución de problemas.

17 – SISTEMAS OPERATIVOS II

Virtualización. Usos. Requerimientos para la virtualización. Tipos de virtualización. Hipervisores de tipo 1. Hipervisores de tipo 2. Paravirtualización. Virtualización de la memoria. Virtualización de la E/S. Dispositivos virtuales Emuladores. Diferencias entre virtualización y emulación. Diferencias entre contenedores y virtualización. Uso de la virtualización y de los contenedores. Conceptos de contenedores: imágenes, tipos de volúmenes, adaptadores de red virtuales. Creación, ejecución y eliminación de contenedores. Archivo de definición de contenedores. Orquestación de contenedores. Concepto. Ejemplos. Programación de Sistemas operativos. Programación en el espacio del usuario y del kernel. Llamadas al sistema (system calls). Desarrollo de controladores de dispositivos.

18 – BASE DE DATOS II

SQL avanzado, ejecución de consultas complejas, uso de subconsultas, funciones agregadas, manejo de joins, vistas e índices. Transacciones y concurrencia en bases de datos, propiedades ACID, control de concurrencia, técnicas de bloqueo y desbloqueo. Administración de bases de datos, gestión de usuarios, asignación de privilegios, realización de copias de seguridad, procedimientos de recuperación ante fallos. Optimización de consultas, técnicas para mejorar el rendimiento de las bases de datos, análisis de planes de ejecución. Bases de Datos Distribuidas. Arquitectura y diseño de bases de datos distribuidas. Replicación y particionamiento de datos. Consistencia, disponibilidad y partición tolerante (CAP Theorem). Protocolos de consenso (Paxos, Raft). Casos de uso y ejemplos prácticos.

Bases de Datos NoSQL. Introducción a bases de datos NoSQL: Conceptos y características. Bases de datos de documentos (MongoDB, CouchDB). Bases de datos de clave-valor (Redis, DynamoDB). Bases de datos de columna (Cassandra, HBase). Bases de datos de grafos (Neo4j). Comparación con bases de datos relacionales y casos de uso.

Optimización y Seguridad de Bases de Datos. Técnicas de optimización: Índices, particionamiento, caché. Optimización de consultas y tuning de bases de datos. Seguridad en bases de datos: Cifrado de datos, control de acceso, auditoría. Recuperación ante desastres y respaldo de datos. Implementación de políticas de seguridad y mejores prácticas.

19 – PRÁCTICA PROFESIONALIZANTE II

Proceso de desarrollo de una solución. Incorporar conceptos de mantenimiento y mejoras como componentes del desarrollo de software. El propósito de estas Prácticas Profesionalizantes de implementación de sistemas en la nube es que el futuro TSDSW proponga una o varias alternativas

que solucionen, mejoren, optimicen e innoven a través de proyectos de desarrollo web. La solución seleccionada deberá plasmarse en documentación técnica de acuerdo a la propuesta.

Los estudiantes, en forma grupal o individual, diseñarán y desarrollarán sistemas en la nube que resuelvan casos o problemáticas específicas. Para ello se deberán determinar las limitaciones de un problema y seleccionar un procedimiento de resolución adecuado. Deberán relevar las necesidades de información del cliente, determinar la mejor solución y desarrollar el sistema en la nube que satisfaga los requerimientos.

Este espacio de práctica pretende afianzar y profundizar los saberes sobre el desarrollo en la nube y la implementación de un sistema web con seguridad, tal que permita la resolución de un problema real, que resuelva un requerimiento de usuario.

Para cumplir este objetivo, las prácticas de este espacio estarán referidas a:

- Integrar y contextualizar los saberes desarrollados en la formación, respondiendo a necesidades que implican el desarrollo de productos específicos.
- Desarrollar de manera integral y coordinada en equipos de trabajo los procedimientos propios del desarrollo de software.
- Integrar su tarea dentro del plan de trabajo del equipo de profesionales/colaboradores.

TERCER AÑO

20 - PROGRAMACIÓN III

Desarrollo de Back-End. Revisión de conocimientos previos: protocolo IP, TCP y UDP, modelo cliente-servidor. RESTful API: Funcionamiento, principios REST, protocolo HTTP, operaciones HTTP y CRUD, formato JSON, códigos de estado HTTP. Framework para API REST: Creación de proyectos usando patrones y buenas prácticas, pruebas de puntos finales, Mapeador Objeto Relacional (ORM), autenticación y seguridad con JWT, paginación, búsqueda y ordenamiento, creación de middleware. Pruebas unitarias y de integración. Despliegue de Aplicaciones: Uso de Git, Docker y Kubernetes.

Frameworks para el Desarrollo Front-End. Tecnologías Web: Protocolos web, principios de ingeniería web, instalación y configuración de herramientas. Lenguajes de Programación: JavaScript y TypeScript, HTML 5, CSS, CSS Flex Box, CSS Grid, Tailwind CSS. Herramientas de Desarrollo: Frameworks para el desarrollo front-end, uso de React, React Hooks, enrutamiento, formularios, comunicación con servicios REST APIs, gestión de estados.

Proyecto Integrador de Programación 3. Integración de Conocimientos: Creación de un proyecto integrador que abarque los contenidos de los bloques anteriores.

21 – ARQUITECTURA Y DISEÑO DE INTERFACES: UI y UX

Introducción a la arquitectura de software. Tipos de arquitectura. Modelo cliente-servidor. APIs REST. Patrones de Arquitectura: Arquitectura en capas (Layered Architecture) Patrón de modelo-

vista-controlador. Arquitectura Web: tecnologías full stack. Arquitectura orientada a eventos (Event-Driven Architecture). Arquitectura limpia (Clean Architecture). Arquitectura sin servidor (Serverless Architecture). Arquitectura de microfrontends. Arquitectura basada en microservicios (Microservices Architecture). El enfoque de los microservicios. Ventajas de los microservicios. Las dificultades de los microservicios. Implementación de microservicios. Estructura de un microservicio. Arquitectura de microservicios. Comunicación entre microservicios: síncrona y asíncrona. Protocolo de Cola de Mensajes Avanzado (AMQP).

Arquitectura para el desarrollo aplicaciones móviles. Arquitectura de aplicación para el aprendizaje de máquinas. Bloque Patrones Arquitectónicos y Diseño de Interfaces. Patrones Arquitectónicos : Data Access Object (DAO). Patrón Repository. Data Transfer Object (DTO). Webhook. Load Balance. Service Registry. Service Discovery. API Gateway, Access token, Single, Circuit Breaker. Patrón de Tuberías y Filtros (Pipes and Filters).

Fundamentos de UX. Principios básicos de usabilidad. Investigación de usuarios y creación de personas. Análisis de escenarios y journeys de usuarios. Principios de UI. Teoría del diseño visual (color, tipografía, iconografía). Diseño de interacción. Diseño responsivo y adaptativo. Sistemas de diseño (design systems). Prototipado y Herramientas. Wireframing y prototipado (uso de herramientas como Figma, Sketch, Adobe XD). Creación de prototipos interactivos. Validación y pruebas de usabilidad. Accesibilidad y Diseño Inclusivo. Principios de accesibilidad web. Normativas y estándares (como WCAG). Inclusión de usuarios con diferentes capacidades. Implementación de UI. Introducción a HTML/CSS y frameworks de UI (Bootstrap, Material Design). Integración con JavaScript para mejorar la experiencia de usuario. Gestión y Metodologías. Colaboración entre diseñadores y desarrolladores. Iteración y mejoras continuas basadas en feedback. Tendencias y Buenas Prácticas. Análisis de casos de estudio. Revisión de tendencias actuales en UI/UX. Evaluación y mejora de interfaces existentes.

22 – AUDITORIA Y CALIDAD DE SISTEMAS

Auditoría operativa o administrativa. Conceptos de sistemas de control interno. Normas. Procedimientos. Ingeniería de procesos. Auditoría informática y de sistemas. Estándares. Herramientas. Software de auditoría. Distintos tipos de auditoría informática. Gestión de calidad. Calidad del software. Factores de la calidad. El costo de la calidad. Control de calidad. Enfoques formales. Calidad y seguridad. Aseguramiento de la calidad. Elementos de aseguramiento de la calidad del software. Enfoques formales. Normas de calidad ISO. Fundamentos de las pruebas del software. Visiones interna y externa de las pruebas. Prueba de caja blanca. Prueba de la estructura de control. Prueba de condición. Prueba de flujo de datos. Prueba de bucle. Pruebas de caja negra. Documentación de prueba y centros de ayuda. Prueba para sistemas de tiempo real. Patrones para pruebas de software.

Distinción entre validación y verificación. Enfoques estáticos y dinámicos. Fundamentos de “testing”, testeo de caja negra y de caja blanca. Pruebas funcionales: generación de casos o datos de prueba,

clases de equivalencia. Pruebas estructurales: pruebas estáticas, pruebas dinámicas, cobertura de la prueba. 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 de seguimiento de ejecución de las pruebas. Herramientas de ejecución de pruebas de acuerdo con el tipo de pruebas. Seguimiento de defectos. Ejecución de pruebas unitarias. Alcance y papel del desarrollador de software. Automatización de pruebas: Introducción y tipos de frameworks de automatización. Modelos de automatización. Emuladores y simuladores. Particularidades en la ejecución de pruebas de performance, pruebas web y de aplicaciones móviles.

23 – CIBERSEGURIDAD

Criptografía. Terminología, objetivos y/o finalidad de la criptografía. Tipos de criptografía. Diferencia entre criptografía y seguridad informática. Limitaciones de la criptografía. La Criptografía como elemento de la seguridad informática. La criptografía en la actualidad. Ventajas y desventajas de la criptografía. Hacker y criptografía. Criptografía y grandes cantidades de datos.

Introducción a la ciberseguridad: Conceptos y terminología. Principios de seguridad de la información (Confidencialidad, Integridad, Disponibilidad). Amenazas y vulnerabilidades comunes. Marco legal y ético en ciberseguridad. Normas y estándares de seguridad (ISO 27001, NIST). Métodos de cifrado: Simétrico y asimétrico. Firmas digitales y certificados. Gestión de claves y criptografía en la nube. Técnicas de autenticación y control de acceso. Bloque Defensa y Respuesta a Incidentes. Estrategias de defensa: Firewalls, sistemas de detección de intrusos (IDS/IPS). Seguridad en redes y sistemas operativos. Análisis forense y respuesta a incidentes. Planificación de la continuidad del negocio y recuperación ante desastres. Herramientas y técnicas para la gestión de incidentes de seguridad.

24 – INGLÉS TÉCNICO III

- **Unidad 1**

Los siguientes descriptores deberán abordarse desde las cuatro macro-habilidades (comprensión lectora y auditiva y producción escrita y oral) y orientarse al campo de desempeño del futuro profesional, desarrollando las competencias básicas fijadas por los estándares del Marco Común Europeo de Referencia para la competencia lingüística de nivel B1: al finalizar el cursado, el estudiante será capaz de comprender los puntos principales de textos claros y en lengua estándar si tratan sobre cuestiones que le son conocidas, ya sea en situaciones de trabajo, de estudio o de ocio. Sabrá desenvolverse en la mayor parte de las situaciones que pueden surgir durante un viaje por zonas donde se utiliza la lengua. Podrá producir textos sencillos y coherentes sobre temas que

le son familiares o en los que tiene un interés personal. Podrá describir experiencias, acontecimientos, deseos y aspiraciones, así como justificar brevemente sus opiniones o explicar sus planes.

Contenidos mínimos:

El estudiante deberá desarrollar habilidades de producción (habla y escritura) aplicadas a situaciones específicas de la carrera y el año que transita.

Pasado perfecto. Pasado perfecto continuo. Discurso indirecto. Verbos modales en estructuras perfectivas. Condicional 3.

- **Unidad 2**

Los siguientes descriptores deberán abordarse desde las cuatro macro-habilidades (comprensión lectora y auditiva y producción escrita y oral) y orientarse al campo de desempeño del futuro profesional, desarrollando las competencias básicas fijadas por los estándares del Marco Común Europeo de Referencia para la competencia lingüística de nivel B2: al finalizar el cursado, el estudiante será capaz de entender las ideas principales de textos complejos que traten de temas tanto concretos como abstractos, incluso si son de carácter técnico siempre que estén dentro de su campo de especialización. Podrá relacionarse con hablantes nativos con un grado suficiente de fluidez y naturalidad de modo que la comunicación se realice sin esfuerzo por parte de ninguno de los interlocutores. Podrá producir textos claros y detallados sobre temas diversos así como defender un punto de vista sobre temas generales indicando los pros y los contras de las distintas opciones.

Contenidos mínimos:

El estudiante deberá desarrollar habilidades de producción (habla y escritura) aplicadas a situaciones específicas de la carrera y el año que transita.

Tiempos verbales simples, continuos, perfectos y perfectos continuos contrastados. Verbos modales y expresiones equivalentes, simples y perfectivos. Condicionales 0, 1, 2 y 3. verbos modales en voz pasiva.

25 – LEGISLACIÓN INFORMÁTICA Y ÉTICA PROFESIONAL

- **Contenidos de legislación informática**

Introducción al derecho informático: Conceptos y terminología. Leyes y regulaciones nacionales e internacionales que afectan la informática. Contratos informáticos: Licencias de software, contratos de servicios y outsourcing. Responsabilidad legal de los profesionales de TI. Ética profesional en informática.

Normativas de protección de datos: GDPR, Ley de Protección de Datos Personales (nacional). Principios de protección de datos: Consentimiento, finalidad, minimización de datos. Derechos de los titulares de datos: Acceso, rectificación, supresión.

Seguridad de la información y medidas técnicas para la protección de datos. Impacto de la protección de datos en el desarrollo de software. Bloque Propiedad Intelectual y Delitos Informáticos. Derechos de autor y software: Licencias, registro y protección de software. Patentes

de software: Procedimientos y controversias. Delitos informáticos: Tipos de delitos y su marco legal. Ciberdelito y su impacto en la sociedad. Prevención y mitigación de delitos informáticos.

- **Contenidos de ética profesional**

Introducción a la Ética Profesional. Definición de ética y moral. Importancia de la ética en la ingeniería de software. Códigos de ética profesionales (ACM, IEEE). Casos históricos de fallos éticos en el desarrollo de software. Responsabilidad del Desarrollador de Software. Derechos y obligaciones de los profesionales del software. Propiedad intelectual y derechos de autor. Plagio y el uso correcto de software y código fuente.

Privacidad y Confidencialidad. Privacidad en la Era Digital. Manejo ético de datos personales y confidenciales. Regulaciones y leyes sobre privacidad (GDPR, CCPA). Casos de violaciones de privacidad y sus consecuencias. Seguridad en el Desarrollo de Software. Principios de diseño seguro. Impacto de la falta de seguridad en la sociedad. Ética en la divulgación de vulnerabilidades. Influencia del Software en la sociedad. Accesibilidad e inclusión digital. Brecha digital y su impacto en diferentes comunidades. Desarrollo de software para el bien social.

Inteligencia Artificial y Ética. Desarrollo y Uso Ético de la IA. Sesgos en algoritmos y su impacto social. Transparencia y responsabilidad en sistemas automatizados. Futuro de la IA y sus implicaciones éticas. Ética en la Colaboración y Comunicación. Comunicación efectiva y ética en el entorno profesional. Resolución de conflictos éticos en el trabajo. Ética en la gestión de proyectos de software. Modelos de Toma de Decisiones Éticas. Análisis de casos de estudio. Desarrollo de un marco personal para la toma de decisiones éticas. Reflexión sobre la responsabilidad social y ética profesional.

26 – ESTADÍSTICA Y PROBABILIDADES PARA EL DESARROLLO DE SOFTWARE

Estadística. Definiciones y conceptos fundamentales. Estadística descriptiva. Análisis descriptivo de datos individuales y agrupados: variables discretas y continuas, medidas de posición y de dispersión, histogramas. Efecto del descarte de datos extremos en las diferentes medidas de posición y dispersión. Variables aleatorias discretas y continuas. Distribuciones discretas y continuas, binomial y de Pascal. Recolección de datos para análisis estadísticos, clasificación e interpretación. La distribución normal. Análisis de varianza. Esperanza matemática, variables aleatorias enteras. Ley de los grandes números.

Teoría de la probabilidad. Cálculo de probabilidades. Probabilidad condicional. Combinatoria. Permutaciones. Variaciones y combinaciones. Suceso aleatorio. Sucesos simples y compuestos. Espacio muestral y espacio de sucesos. Propiedades de la probabilidad. Probabilidad total. Probabilidad compuesta. Probabilidad condicionada. Sucesos independientes. Sucesos dependientes. Modelos Probabilísticos. Espacio finito de probabilidades, medida de probabilidad, eventos. Probabilidad condicional, independencia, teorema de Bayes.

27 – GESTIÓN DE PROYECTOS DE SOFTWARE

Concepto de proyecto. Elementos de Gestión. Etapas y criterios para la planificación de Proyectos. Campos de aplicación. La producción por proyectos. Los proyectos en las organizaciones. Enfoque de gestión de proyectos. Etapas en la gestión de un proyecto. Métodos de planificación de proyectos. Métodos PERT/CPM. Diagramas de redes. Concepto de camino crítico. Diagramas temporales de planificación de proyectos. Diagramas de Gantt.

Planeamiento, gestión y control. El planeamiento: Concepto. Proceso de planeamiento, determinación de objetivos, análisis, evaluación y selección de alternativas. Gestión de calidad: normativas. Prevención de riesgos laborales, condiciones y medio ambiente de trabajo. Cuidado del ambiente, eficiencia energética y uso racional de los recursos naturales.

Organización. Tipos, estructura y organigrama. Gestión de los recursos humanos. Trabajo colaborativo. Habilidades para el trabajo en equipo. Coordinación de tareas. Vinculación con el usuario. Relevamiento. Técnicas de relevamiento. Análisis de requisitos. Clasificación de los requerimientos en imprescindibles y deseables. Comunicación con el usuario. Presentación. Negociación y acuerdos relativos al alcance del proyecto. Ejercicio legal de la profesión. Normativa vigente. Responsabilidad y compromiso social.

28 – METODOLOGÍAS ÁGILES

Cómo evolucionó la forma de planificar y ejecutar un proyecto a lo largo del tiempo. Cómo trabajar en entornos complejos y que viene a resolver Scrum. Introducción a los métodos ágiles. Principal problemática de gestión de proyectos. Cómo comprender y gestionar un contexto complejo. Contexto iterativo e incremental de trabajo. Scrum Resolución. Un marco de trabajo ágil: principios y valores. Desarrollo de productos innovadores y creativos. Manifiesto Ágil. Elementos constitutivos de Scrum, eventos. El control empírico del proceso. Pilares: transparencia, inspección y adaptación continua. Los roles en Scrum, Dueño del Producto, Scrum Master y equipo. Las dimensiones de la estrategia, táctica y operativa. De la estrategia al software funcional. El aporte de Lean, XP y Kanban al marco. De los requerimientos a las historias de usuario. Visión del Producto e Historias de Usuario. Concepto del Mínimo Producto Viable. Estimación relativa de requerimientos. Internalización de Scrum mediante deconstrucción. Construcción incremental e iterativa de un producto. Condiciones del inicio y finalización. Tabique y flujos de trabajo. Conceptos de INVEST y MOSCOW. Releases y Product Backlog. La planificación del sprint 0. Técnicas retrospectivas para Sprint Review y Reuniones Retrospectivas. Simulación de desarrollo de un producto.

29 – BASE DE DATOS III

Bases de Datos Distribuidas: Arquitectura, replicación de datos, fragmentación, desafíos de consistencia de datos. Data Warehousing: Diseño de almacenes de datos, procesos ETL (Extract, Transform, Load), uso de OLAP (Online Analytical Processing) para análisis. Big Data y Bases de Datos NoSQL: Conceptos clave de Big Data, profundización en bases de datos NoSQL, casos de

uso en la industria. Bases de Datos No Relacionales: Tipos y características de bases de datos no relacionales, comparación con bases de datos relacionales tradicionales.

Bases de Datos Documentales: Estructura y características de bases de datos documentales, almacenamiento y recuperación de documentos en formatos como JSON y BSON. Bases de Datos Orientadas a Objetos: Modelado y gestión de datos en bases de datos orientadas a objetos, mapeo de objetos a bases de datos. Bases de Datos Vectoriales: Conceptos y aplicaciones de bases de datos vectoriales, uso en análisis de datos espaciales y geoespaciales.

Seguridad en Bases de Datos: Protección de datos mediante cifrado, autenticación y autorización de usuarios, implementación de auditorías de seguridad. Proyecto Integrador: Aplicación de conocimientos adquiridos en los tres niveles para resolver problemas prácticos, desarrollo de soluciones completas en entornos simulados o reales.

30 – PRÁCTICA PROFESIONALIZANTE III

En el último espacio de Práctica Profesionalizante, los estudiantes deberán elaborar un proyecto de carácter integrador que contemple todos los aprendizajes adquiridos previamente y su transferencia a un recorte concreto de la realidad. El objetivo principal de este trabajo es proponer alternativas tecnológicas que optimicen y/o innoven en relación a problemáticas sociales y/o específicas de su profesión, desarrollando e implementando sistemas web, móviles u otro tipo de acuerdo a requerimientos de usuarios.

Por su carácter integral, estas prácticas profesionalizantes requieren poner en juego diversas actividades propias del quehacer profesional. Para ello los estudiantes, en forma grupal o individual, irán tomando una serie de decisiones y realizando un conjunto de actividades que les permita llegar al objetivo final.

Este espacio curricular está organizado por un taller grupal y el trabajo en campo. En el taller, los estudiantes preparan su salida al campo, pondrán en común sus vivencias y realizarán la síntesis y conclusiones de las experiencias. El trabajo en campo corresponderá a la inserción de los estudiantes en el contexto laboral o en aquel que mejor se aproxime a las condiciones reales. La estrategia de evaluación final consistirá en la presentación y defensa del proyecto.

En la planificación de esta unidad curricular, deberán preverse un número suficiente de encuentros presenciales para desarrollar los requisitos y criterios que deberán tener en cuenta los estudiantes para la elaboración del proyecto, a través de materiales y guías de trabajo. En las clases presenciales los estudiantes planifican su trabajo y consultan dudas frente a la elaboración del proyecto. Se socializan las experiencias que van adquiriendo, se gestionan los permisos y seguros que se requieran para realizar visitas en contextos de trabajo, el uso del laboratorio de programación para la elaboración del trabajo. El objetivo principal de los encuentros será la reflexión grupal sobre las dudas y los avances con el fin de lograr un ámbito de aprendizaje colaborativo.

De acuerdo con la planificación y formato de prácticas profesionalizantes adoptado por la institución, la evaluación de esta PP III involucrar una o más instancias que posibiliten evaluar las distintas

dimensiones del proyecto integrador; entre ellas la pertinencia y factibilidad técnica de la propuesta, su fundamentación documentada, los desempeños de los estudiantes en el proceso de elaboración del proyecto, etc. Algunos criterios de evaluación de particular relevancia para estas PP III son:

- Coherencia en la sistematización y estructuración del proyecto.
- Fundamentación de la propuesta
- Síntesis argumental pertinente teniendo en cuenta los saberes adquiridos a lo largo de su trayectoria educativa.
- Exposición autónoma con capacidad de oratoria y defensa fundamentada y consistente de ideas.
- Honestidad intelectual. Responsabilidad ético - legal y social.
- Manejo adecuado del vocabulario científico y técnico de su campo laboral

El desarrollo del Proyecto Integrador comprende una serie de encuentros grupales presenciales en los cuales los estudiantes planifican su trabajo, socializan las experiencias que van adquiriendo en cada una de las etapas de los diversos proyectos y elaboran conclusiones que permitan aprendizajes colectivos.

El docente y las autoridades de la institución educativa acordarán los ámbitos destinados a la realización de las prácticas externas de los estudiantes para el desarrollo del proyecto. Los contextos deberán ser diversos y pertinentes en relación a los objetivos planteados, incluyendo todos los requerimientos para que las actividades prácticas se realicen en contextos seguros.

Prácticas profesionalizantes: Es un espacio de realización de actividades ligadas al campo real de trabajo que tienen como propósito la aproximación progresiva a las múltiples tareas que constituyen el desempeño profesional, a partir de actividades de reflexión y acción. Su desarrollo debe ser progresivo a lo largo de toda la carrera y debe contemplar el análisis permanente de la experiencia que se desarrolla en el contexto de trabajo a través, por ejemplo, de talleres de análisis y reflexión paralelos al desarrollo de la práctica externa, cuya principal finalidad es facilitar procesos que promuevan en los estudiantes un desempeño profesional idóneo y éticamente orientado. Si bien las prácticas profesionalizantes pueden asumir diferentes formatos, sintéticamente pueden reducirse a dos: a) pasantías, prácticas en ambientes de trabajo y/ o formación en ambientes productivos (reales y/o simulados), b) proyectos orientados a la resolución de problemáticas y necesidades a nivel institucional/organizacional, local y/ o regional. En todos los casos, las prácticas profesionalizantes son organizadas y coordinadas por la institución educativa a través de los entornos formativos existentes y los acuerdos que se generen para la realización por fuera de ella. Cualquiera sea el formato adoptado, los IES tienen la responsabilidad de planificar, organizar, supervisar y evaluar las prácticas profesionalizantes. La planificación y desarrollo de esta unidad curricular deberá asegurar, de manera continua a lo largo de cada cuatrimestre, instancias presenciales a cargo del equipo docente, las cuales tendrán como actividad específica la realización del seguimiento de las distintas actividades que las/os estudiantes deberán efectuar en función de las características de cada una de las prácticas profesionalizantes establecidas en el diseño

curricular.

5. RÉGIMEN DE REGULARIDAD, PROMOCIÓN, EVALUACIÓN Y ACREDITACIÓN. Cfr. RESOLUCIÓN RAM VIGENTE.

Cfr. RESOLUCIÓN 1286-DGE-2024 RAM, RAI y demás normativas vigentes

5.1. Espacios Curriculares acreditables por formación previa

Las acreditaciones se realizarán cuando haya certificación de organismos oficiales (DGE, Min. de Educación, Min de Trabajo, entre otros) o por evaluación de idoneidad a cargo del Instituto de Educación Superior, junto con el sector socio – productivo correspondiente

5.2. Espacios curriculares de acreditación directa

Se regirán por el sistema de acreditación directa, según lo dispone el Artículo 46 inciso "a" del apartado referido a las trayectorias estudiantiles correspondiente al Reglamento Académico Marco (Res. N° 1286-DGE-2024), los siguientes espacios curriculares:

6. RÉGIMEN DE CORRELATIVIDADES*

ESPACIO CURRICULAR	REGULAR PARA CURSAR	APROBADA PARA ACREDITAR
PRIMER AÑO		
10 – PRÁCTICA PROFESIONALIZANTE I	-----	PROGRAMACIÓN I BASE DE DATOS I
SEGUNDO AÑO		
11 – PROGRAMACIÓN II	PROGRAMACIÓN I	PROGRAMACIÓN I
12 – MATEMÁTICA DISCRETA	MATEMÁTICA APLICADA	MATEMÁTICA APLICADA
13 – ANÁLISIS MATEMÁTICO	MATEMÁTICA DISCRETA	MATEMÁTICA DISCRETA
14 – COMUNICACIÓN Y REDES	ARQUITECTURA DE DISPOSITIVOS	ARQUITECTURA DE DISPOSITIVOS
15 – INGLÉS TÉCNICO II	INGLÉS TÉCNICO I	INGLÉS TÉCNICO I
16 – MODELADO DE SOFTWARE	LÓGICA COMPUTACIONAL ALFABETIZACIÓN ACADÉMICA	PROGRAMACIÓN I LÓGICA COMPUTACIONAL ALFABETIZACIÓN ACADÉMICA
17 – SISTEMAS OPERATIVOS II	SISTEMAS OPERATIVOS I	SISTEMAS OPERATIVOS I
18 – BASE DE DATOS II	BASE DE DATOS I	BASE DE DATOS I
19 – PRÁCTICA PROFESIONALIZANTE II	PRÁCTICA PROFESIONALIZANTE I	PRÁCTICA PROFESIONALIZANTE I
TERCER AÑO		
20 – PROGRAMACIÓN III	PROGRAMACIÓN II	PROGRAMACIÓN II
21 – ARQUITECTURA Y DISEÑO DE INTERFACES: UI Y UX	MODELADO DE SOFTWARE	MODELADO DE SOFTWARE
22 – AUDITORÍA Y CALIDAD DE SISTEMAS	MODELADO DE SOFTWARE	MODELADO DE SOFTWARE
23 – CIBERSEGURIDAD	-----	LEGISLACIÓN INFORMÁTICA Y ÉTICA PROFESIONAL
24 – INGLÉS TÉCNICO III	INGLÉS TÉCNICO II	INGLÉS TÉCNICO II
25 – LEGISLACIÓN INFORMÁTICA Y ÉTICA PROFESIONAL	-----	-----
26 – ESTADÍSTICA Y PROBABILIDADES PARA EL DESARROLLO DE SOFTWARE	ANÁLISIS MATEMÁTICO	ANÁLISIS MATEMÁTICO
27 – GESTIÓN DE PROYECTOS DE SOFTWARE	MODELADO DE SOFTWARE	MODELADO DE SOFTWARE

28 – METODOLOGÍAS ÁGILES	---	GESTIÓN DE PROYECTOS DE SOFTWARE
29 – BASE DE DATOS III	BASE DE DATOS II	BASE DE DATOS II
30 – PRÁCTICA PROFESIONALIZANTE III	PRÁCTICA PROFESIONALIZANTE II	PRÁCTICA PROFESIONALIZANTE II

***Para poder cursar tercer año el estudiante deberá tener acreditado todos los espacios curriculares de primer año y la mitad más uno de los espacios curriculares de segundo año.**

5. RÉGIMEN DE ASISTENCIA

De acuerdo a los formatos curriculares y proyecto de modalidad combinada aprobado por DES y en el marco de lo establecido por la RES.1286-DGE-2024 RAM.

III. IMPLEMENTACIÓN DE LA CARRERA

1. Recursos

1.1. Humanos

Se seleccionarán aquellos docentes que cumplan los requisitos previstos en la normativa específica sobre el ingreso y/o reasignación de docentes correspondiente al Nivel Superior jurisdiccional (Decreto Ley N° 530/18 – Cap. III) y la Ley de Educación Provincial N° 6970 Título V- De la Educación Superior, Capítulo IV- Gobierno de la Educación Superior no Universitaria, Art. 112°, inc. c)

Perfiles docentes necesarios para cubrir los espacios curriculares:

El perfil profesional docente establecido para cada espacio curricular debe ser considerado prioritario al momento de asignar las horas de este. Se priorizará a los postulantes con formación docente acreditable, con conocimiento, posgrado, antecedentes y experiencia en el campo (Decreto 530-DGE-2018). En los espacios curriculares de práctica profesionalizante, la experiencia laboral en el campo de formación es excluyente. Los espacios con sus perfiles docentes correspondientes son los siguientes.

ESPACIO CURRICULAR	PERFIL PROFESIONAL
1. PROGRAMACIÓN I	Licenciado en Informática, Ingeniero en Sistemas/ Informática/ Computación.
2. MATEMÁTICA APLICADA	Licenciado /Profesor de Matemática. Ingeniero en sistemas de información.
3 – LÓGICA COMPUTACIONAL	Profesor y/o licenciado en Matemática. Licenciado y/o ingeniero en Sistemas/ Informática/ Computación.
4 – INGLÉS TÉCNICO I	Profesor/Licenciado en Inglés.
5 – ALFABETIZACIÓN ACADÉMICA	Licenciado / Prof. en Lengua y Literatura con postitulación en TIC.
6 – SEMINARIO NUEVAS TECNOLOGÍAS	Licenciado y/o Ingeniero en Sistemas/ Informática/ Computación.
7 – ARQUITECTURA DE DISPOSITIVOS	Licenciado en Informática, Ingeniero en Sistemas/ Informática/ Computación.
8 – SISTEMAS OPERATIVOS I	Licenciado y/o Ingeniero en Sistemas/ Informática/ Computación.
9 – BASE DE DATOS I	Licenciado y/o Ingeniero en Sistemas/ Informática/ Computación.
10 – PRÁCTICA PROFESIONALIZANTE I	Licenciado y/o Ingeniero en Sistemas/Informática/ Computación con formación y experiencia pedagógica.
11 – PROGRAMACIÓN II	Licenciado en Informática, Ingeniero en Sistemas/ Informática/ Computación
12 – MATEMÁTICA DISCRETA	Licenciado /Profesor de Matemática. Ingeniero en sistemas de información.
13 – ANÁLISIS MATEMÁTICO	Licenciado /Profesor de Matemática. Ingeniero en sistemas de información.
14 – COMUNICACIÓN Y REDES	Licenciado y/o Ingeniero en Sistemas/ Informática/ Computación, Ingeniero en Telecomunicaciones/ Comunicación/ Redes.
15 – INGLÉS TÉCNICO II	Profesor/Licenciado en Inglés.
16 – MODELADO DE SOFTWARE	Licenciado y/o Ingeniero en Sistemas/Informática/ Computación.
17 – SISTEMAS OPERATIVOS II	Licenciado y/o Ingeniero en Sistemas/ Informática/ Computación.

18 – BASE DE DATOS II	Licenciado y/o Ingeniero en Sistemas/ Informática/ Computación.
19 – PRÁCTICA PROFESIONALIZANTE II	Licenciado y/o Ingeniero en Sistemas/Informática/ Computación con experiencia en programación y formación y experiencia pedagógica.
20 – PROGRAMACIÓN III	Licenciado en Informática, Ingeniero en Sistemas/ Informática/ Computación
21 – ARQUITECTURA Y DISEÑO DE INTERFACES: UI Y UX	Licenciado y/o Ingeniero en Sistemas/Informática/ Computación.
22 – AUDITORÍA Y CALIDAD DE SISTEMAS	Licenciado y/o Ingeniero en Sistemas/ Informática/ Computación.
23 – CIBERSEGURIDAD	Licenciado y/o Ingeniero en Sistemas/ Informática/ Computación.
24 – INGLÉS TÉCNICO III	Profesor/Licenciado en Inglés
25- LEGISLACIÓN INFORMÁTICA Y ÉTICA PROFESIONAL	Profesor/ Licenciado en Ciencias Jurídicas Contables. Abogado con antecedentes en el área de sistemas.
26 – ESTADÍSTICA Y PROBABILIDADES PARA EL DESARROLLO DE SOFTWARE	Licenciado/Profesor en Matemática. Ingeniero en Sistemas, Ing. o Lic en Informática.
27 – GESTIÓN DE PROYECTOS DE SOFTWARE	Licenciado y/o Ingeniero en Sistemas/Informática/ Computación.
28 – METODOLOGÍAS ÁGILES	Licenciado y/o Ingeniero en Sistemas/Informática/ Computación.
29 – BASE DE DATOS III	Licenciado y/o Ingeniero en Sistemas/ Informática/ Computación.
30 – PRÁCTICA PROFESIONALIZANTE III	Licenciado y/o Ingeniero en Sistemas/Informática/ Computación con experiencia en programación y formación y experiencia pedagógica.

Entorno Formativo en el lugar dónde se dictará la propuesta formativa.

- **Condiciones edilicias**

Edificios propios del IES o en edificios compartidos con otras entidades de educación oficial o privada.

- **Equipamientos**

Proyectores multimedia, Equipos de informática portátiles, Herramientas de uso cotidiano. Elementos didácticos para la enseñanza de materias relacionadas con la informática (*Hardware*

diverso, elementos activos de red, Instaladores de software, etc.)

Medios accesibles para traslado para la práctica en el campo laboral

- **Instalaciones**

Aulas, Salas de Informática, Servicios de Internet

- **Ámbitos de prácticas**

Se mantendrán e incrementarán convenios con empresas y organismos del sector socio productivo dentro de los ámbitos públicos y privados. Cfr. Resolución N° 2992-DES-15 “Acuerdo Marco de Prácticas Profesionalizantes” y Reglamentos Institucionales de Prácticas Profesionalizantes. Cobertura de seguro para docentes y estudiantes en salidas de campo.

Aulas, laboratorios propios, convenios con empresas.

- **Bibliotecas técnicas especializadas**

Bibliotecas propias. Biblioteca específica de la carrera. Bibliotecas virtuales. Indispensable el acceso a Internet

CURSO DE INGRESO

Se sugiere que la duración de este curso no debe ser inferior a tres meses. Los institutos arbitrarán la modalidad del cursado, de acuerdo a la Res. N° 1286- DGE-2024 (RAM) y RAI vigentes.

Se sugiere implementar en los cursos nivelatorios de ingreso el abordaje de contenidos de **Lógica e Introducción a la Programación.**