

CÓRDOBA, 6 de diciembre de 2024.-

VISTO:

La propuesta formativa de pregrado referida a la programación en todas sus etapas dentro del ámbito del Instituto de Gestión e Innovación Tecnológica y Productiva de la Universidad Provincial de Córdoba;

CONSIDERANDO:

Que desde el Instituto de Gestión e Innovación Tecnológica y Productiva de la Universidad Provincial de Córdoba se eleva una propuesta formativa de pregrado relativa a la programación en todas sus etapas ("front-end" como en el "back-end") con formato de tecnicatura.

Que conforme se fundamenta, el desarrollo de software se ha convertido en uno de los pilares fundamentales de la economía digital y la innovación en múltiples sectores. La demanda de profesionales con habilidades en programación "full stack" (capaces de trabajar tanto en el "front-end" como en el "back-end" de las aplicaciones) ha experimentado un notable incremento en todo el mundo, impulsada por la necesidad de soluciones tecnológicas integrales y personalizadas.

Que la Universidad Provincial de Córdoba busca dar una respuesta a las tendencias y necesidades actuales del mercado laboral, a la demanda de innovación tecnológica y a la necesidad de contar con profesionales formados en áreas estratégicas de desarrollo de software.

Que dicha tecnicatura no solo brindará oportunidades de inserción laboral rápida y de calidad a sus egresados, sino que también contribuirá al crecimiento y competitividad de la industria tecnológica a nivel local y nacional, fortaleciendo la economía del conocimiento y posicionando a la región como un centro de talento calificado en tecnología.

Que para obtener el título de “Técnico/a Universitario/a en Programación Full Stack” los/as estudiantes deberán aprobar todas las unidades curriculares establecidas en el plan de estudios, además deberán aprobar un examen de suficiencia de idioma inglés.

Que la oferta cuenta con el Visto Bueno de la Secretaría Académica y de Posgrado de la Universidad.

Que conforme lo establece el inc. C) del artículo 22 del Estatuto Universitario aprobado por Resolución Rectoral Nro. 173/2024 corresponde al Consejo superior “... crear, modificar y/o extinguir Ciclos de Complementación, Carreras...”.

Que ante ello corresponde aprobar la propuesta de formación de pregrado y crear la “Tecnicatura Universitaria en Programación Full Stack” elevada por el Instituto de Gestión e Innovación Tecnológica y Productiva de la Universidad Provincial de Córdoba.

Que conforme a lo dispuesto por el art. 14 de la Ley Provincial Nro. 9.375, su modificatoria Ley Provincial Nro. 10.206, el Decreto Nro. 1.080/18, la Ley Provincial Nro. 10.704, la Resolución del Ministerio de Educación Nro. 591 - Letra D/2024, la Ley Provincial Nro. 10.953 y demás normativa aplicable, corresponden a la Rectora Normalizadora las atribuciones propias de su cargo y a su vez aquellas que el Estatuto les asigna a los futuros órganos de gobierno de la Universidad.

En virtud de todo ello, normativa citada y en usos de sus atribuciones;

**LA RECTORA NORMALIZADORA
DE LA UNIVERSIDAD PROVINCIAL DE CÓRDOBA
RESUELVE:**

ARTÍCULO 1º: **CRÉASE** bajo el ámbito del Instituto de Gestión e Innovación Tecnológica y Productiva dependiente de la Universidad Provincial de Córdoba, la carrera universitaria de pregrado: “Tecnicatura Universitaria en Programación Full Stack”, con una duración de tres (3) años y con una carga horaria de un mil seiscientos noventa y seis (1696) horas reloj, la cual otorga el título de: “Técnico/a Universitario/a en Programación Full Stack”.

ARTÍCULO 2º: **APRUÉBASE** el plan de estudios que en Anexo se acompaña y forma parte de la presente Resolución.

ARTÍCULO 3º: **PROTOCOLÍCESE**, comuníquese y archívese

RESOLUCIÓN Nro. 0479.-




Esp. María Julia Oliva Cúneo
Rectora Normalizadora
Universidad Provincial de Córdoba

ANEXO

UNIVERSIDAD PROVINCIAL DE CÓRDOBA
INSTITUTO DE GESTIÓN E INNOVACIÓN TECNOLÓGICA Y PRODUCTIVA
TECNICATURA UNIVERSITARIA EN PROGRAMACIÓN FULL STACK

1. Identificación de la carrera

1.1. Nombre de la Carrera

Tecnicatura Universitaria en Programación Full Stack

1.2. Nombre del título a otorgar

Técnico/a Universitario/a en Programación Full Stack

1.3. Duración estimada

3 (tres) años

1.4. Carga horaria total

1696 (un mil seiscientos noventa y seis) horas reloj

1.5. Nivel académico universitario

Pregrado

1.6. Modalidad

Presencial

1.7. Ubicación en la estructura institucional

Instituto de Gestión e Innovación Tecnológica y Productiva, Universidad Provincial de Córdoba

1.8. Fundamentación

En la actualidad, el desarrollo de software se ha convertido en uno de los pilares fundamentales de la economía digital y la innovación en múltiples sectores. La demanda de profesionales con habilidades en programación "full stack" (capaces de trabajar tanto en el "front-end" como en el "back-end" de las aplicaciones) ha experimentado un notable

incremento en todo el mundo, impulsada por la necesidad de soluciones tecnológicas integrales y personalizadas.

Las organizaciones (públicas y privadas) buscan profesionales que puedan gestionar todas las etapas de un proyecto y los programadores full stack se constituyen como profesionales referentes por su capacidad para entender y contribuir en su desarrollo con una mirada 360°. Al tener conocimientos en ambas áreas puede comunicarse más efectivamente con los miembros del equipo como diseñadores y desarrolladores, además de poder interpretar las necesidades reales de su cliente. Esto también les permite a los programadores poder asumir roles de liderazgo y de gestión de proyectos. La creación de una Tecnicatura Universitaria en Programación Full Stack responde a esta creciente demanda y tiene como objetivo formar profesionales con competencias sólidas y adaptables, que puedan insertarse rápidamente en el mercado laboral y contribuir al crecimiento de las industrias basadas en tecnología.

Con la incorporación de esta Tecnicatura Universitaria, la UPC busca dar una respuesta a las tendencias y necesidades actuales del mercado laboral, a la demanda de innovación tecnológica y a la necesidad de contar con profesionales formados en áreas estratégicas de desarrollo de software. Esta tecnicatura no solo brindará oportunidades de inserción laboral rápida y de calidad a sus egresados, sino que también contribuirá al crecimiento y competitividad de la industria tecnológica a nivel local y nacional, fortaleciendo la economía del conocimiento y posicionando a la región como un centro de talento calificado en tecnología.

Los graduados podrán integrarse en equipos de trabajo de empresas tecnológicas, emprender proyectos propios o desempeñarse como freelancers, impulsando el desarrollo tecnológico y económico en sus comunidades, lo cual generará un impacto positivo en las economías locales, provincial y nacional contribuyendo a la innovación y competitividad en diferentes sectores productivos.

2. Horizontes de la carrera

2.1. Objetivos de la carrera

- Formar profesionales con habilidades técnicas para la programación full stack, colaborar en la creación, implementación y el mantenimiento de aplicaciones y sistemas de software de manera integral, desde el front-end hasta el back-end, optimizando la experiencia de usuario y proporcionando conocimientos en lenguajes, frameworks y bases de datos SQL y NoSQL, contribuyendo, de esta manera, al

desarrollo de soluciones tecnológicas innovadoras y a la mejora de procesos en organizaciones de diversos sectores.

- Facilitar el desarrollo de habilidades para la resolución de problemas y el pensamiento crítico, mediante la práctica de la programación y el análisis de problemas complejos en proyectos de software, así como de las habilidades de diseño de soluciones para que, con su aporte, sean funcionales, escalables y orientadas a la experiencia de usuario.
- Desarrollar, en los futuros profesionales, habilidades para ayudar a lograr adecuadas integraciones que aseguren un flujo de datos eficiente y seguro entre el cliente y el servidor.
- Capacitar a los estudiantes en metodologías ágiles que permitan participar en el desarrollo de proyectos de manera iterativa, colaborativa y eficientes, promoviendo la entrega continua de valor y la adaptación al cambio, facilitando experiencias de aprendizaje colaborativo mediante proyectos en equipo, donde puedan aplicar y afianzar los principios de desarrollo ágil en un entorno de simulación de proyectos reales.
- Desarrollar habilidades en el uso de sistemas de control de versiones, fundamentales para la colaboración y el trabajo en equipo en entornos de desarrollo, en prácticas de documentación de código, pruebas automatizadas, y en el uso de entornos de desarrollo colaborativo que aseguren la calidad y sostenibilidad de los proyectos de software.
- Promover el aprendizaje de seguridad en la programación de software, integrando principios de seguridad informática en la programación de aplicaciones full stack, de modo que puedan colaborar en lo relativo a la protección de la integridad y confidencialidad de los datos, así como en la adopción de medidas preventivas para protegerse de ciberataques.

2.2 Perfil de la persona egresada

Se espera que al finalizar su proceso de formación el/a *Técnico/a Universitario/a en Programación Full Stack* haya logrado adquirir conocimientos, habilidades/competencias y actitudes para:

- La realización de desarrollos full stack, tanto en la programación con la mirada del cliente (front-end) como del servidor (back-end), con conocimientos en tecnologías de desarrollo, frameworks y bases de datos, entre otras

- La administración de servidores, seguridad de la información y desarrollo ágil, esenciales para crear aplicaciones robustas, seguras y escalables.
- El trabajo en equipo en proyectos integrales y de manera colaborativa en todas las fases del desarrollo de un proyecto.
- La utilización de metodologías ágiles en entornos de desarrollo dinámicos y centrados en la entrega continua de valor.

2.3. Alcances del título

El/a Técnico/a Universitario/a en Programación Full Stack está capacitado/a para desempeñar las siguientes actividades laborales:

- Programar, junto a otros, aplicaciones complejas, creando y manteniendo aplicaciones web y móviles, asegurando su funcionalidad y rendimiento en diversas plataformas.
- Participar en la gestión de bases de datos, implementando y administrando las mismas en SQL y NoSQL, garantizando la integridad y seguridad de la información.
- Aplicar metodologías ágiles para la gestión de proyectos, promoviendo la adaptabilidad y la entrega continua de valor.
- Trabajar en equipos colaborativos, participando activamente en entornos profesionales.
- Implementar nuevas tecnologías, manteniendo un aprendizaje continuo y adaptándose a las tendencias emergentes en la programación de software.
- Participar de procesos de innovación, contribuyendo a proyectos que promuevan la mejora tecnológica en diversos sectores productivos, incorporando principios de seguridad informática en el ciclo de vida del desarrollo, creando aplicaciones robustas y seguras.
- Crear e implementar emprendimientos independientes en el ámbito del software y la tecnología

3. Diseño curricular de la carrera

3.1. Requisitos de ingreso

En virtud de lo establecido en el artículo 7 de la Ley de Educación Superior 24521/95, para ingresar a instituciones de la Educación Superior el ingresante debe tener completos sus estudios secundarios. También se prevé que las personas "mayores de

25 años que no reúnan esa condición, podrán ingresar siempre que demuestren, a través de las evaluaciones que en su caso establezcan, que tienen preparación y/o experiencia laboral acorde con los estudios que se proponen iniciar, así como aptitudes y conocimientos suficientes para cursarlos satisfactoriamente”.¹

3.2. Requisitos de egreso

Para obtener el título de Técnico/a Universitario/a en Programación Full Stack los/as estudiantes deberán aprobar todas las unidades curriculares establecidas en el plan de estudios. Además, deberán aprobar un examen de suficiencia de idioma inglés.

3.3. Estructura curricular

a. Unidades Curriculares, código, formato, asignación horaria semanal, total y condición académica

¹ Ley de Educación Superior 24521/95. Art. 7 del Capítulo 2: De la Estructura y articulación de la Educación Superior.

| Primer Año | | | | | | | | | | | |
|------------------------------------|--------|------------------|--------------------|-------------------------|---|-------------|------------------|--------------------|-------------------------|-------------------|--------|
| Unidades curriculares semestrales | | | | | | | | | | | |
| 1° semestre | | | | | | 2° semestre | | | | | |
| Unidad curricular (UC) | Cód UC | Hs reloj anuales | Hs reloj semanales | Condición académica | Unidad curricular | Cód UC | Hs reloj anuales | Hs reloj semanales | Condición académica | Unidad curricular | Cód UC |
| Introducción al Análisis Sistemico | 01 | 32 | 2 | Promoción Regular Libre | Arquitectura de Computadoras | 06 | 64 | 4 | Promoción Regular Libre | | |
| Base de Datos 1 | 02 | 96 | 6 | Promoción Regular Libre | Base de Datos 2 | 07 | 96 | 6 | Promoción Regular Libre | | |
| Emprendedurismo | 03 | 32 | 2 | Promoción Regular Libre | Metodologías de Resolución de Problemas | 08 | 32 | 2 | Promoción Regular Libre | | |
| Alfabetización Digital | 04 | 32 | 2 | Promoción Regular Libre | Programación 1 | 09 | 96 | 6 | Promoción Regular Libre | | |
| Algoritmos 1 | 05 | 96 | 6 | Promoción Regular Libre | Diseños y Arquitectura de Despliegue 1 | 10 | 32 | 2 | Promoción Regular Libre | | |

Totales 1° año

Unidades curriculares: Diez (10) semestrales

Horas reloj anuales: seiscientas ocho (608)

Horas reloj semanales: primer semestre dieciocho (18) y segundo semestre veinte (20)

| Segundo Año | | | | | | | | | | | |
|--|---------|------------------|--------------------|-------------------------|--|-------------|------------------|--------------------|-------------------------|-------------------|--------|
| Unidades curriculares semestrales | | | | | | | | | | | |
| 1° semestre | | | | | | 2° semestre | | | | | |
| Unidad curricular | Cód. UC | Hs reloj anuales | Hs reloj semanales | Condición académica | Unidad curricular | Cód UC | Hs reloj anuales | Hs reloj semanales | Condición académica | Unidad curricular | Cód UC |
| Diseños y Arquitectura de Despliegue 2 | 11 | 32 | 2 | Promoción Regular Libre | Arquitectura de Internet y APIs REST | 16 | 48 | 3 | Promoción Regular Libre | | |
| Programación Orientada a Objetos | 12 | 96 | 6 | Promoción Regular Libre | Programación Extrema y Desarrollo Guiado por Pruebas | 17 | 48 | 3 | Promoción Regular Libre | | |
| Base de Datos 3 | 13 | 96 | 6 | Promoción Regular Libre | Sistemas Operativos | 18 | 48 | 3 | Promoción Regular Libre | | |
| Competencias Comunicacionales 1 | 14 | 32 | 2 | Promoción Regular Libre | Competencias Comunicacionales 2 | 19 | 32 | 2 | Promoción Regular Libre | | |
| Programación Web | 15 | 48 | 3 | Promoción Regular Libre | Programación Backend | 20 | 96 | 6 | Promoción Regular Libre | | |

Totales 2° año:

Unidades curriculares: diez (10) semestrales.

Horas reloj anuales: quinientas setenta y seis (576)

Horas reloj semanales: primer semestre diecinueve (19) y segundo semestre diecisiete (17)

0479

| Tercer Año | | | | | | | | | |
|--|---------|------------------|--------------------|-------------------------|--|---------|------------------|--------------------|-------------------------|
| Unidades curriculares semestrales | | | | | | | | | |
| 1° semestre | | | | | 2° semestre | | | | |
| Unidad curricular | Cód. UC | Hs reloj anuales | Hs reloj semanales | Condición académica | Unidad curricular | Cód. UC | Hs reloj anuales | Hs reloj semanales | Condición académica |
| Gestión de Proyectos | 21 | 48 | 3 | Promoción Regular Libre | Introducción a DevOps | 25 | 48 | 3 | Promoción Regular Libre |
| Desarrollo para Dispositivos Móviles y IoT | 22 | 64 | 4 | Promoción Regular Libre | Verificación y Validación de Programas | 26 | 48 | 3 | Promoción Regular Libre |
| Base de Datos 4 | 23 | 96 | 6 | Promoción Regular Libre | Estadística y Ciencia de Datos | 27 | 48 | 3 | Promoción Regular Libre |
| Habilidades para la Gestión | 24 | 32 | 2 | Promoción Regular Libre | Práctica Profesionalizante 1 | 28 | 96 | 6 | Promoción Regular Libre |
| | | | | | Ética y Deontología | 29 | 32 | 2 | Promoción Regular Libre |

Totales 3° año:

Unidades curriculares: nueve (9) semestrales

Horas reloj anuales: quinientas doce (512)

Horas reloj semanales: primer semestre quince (15) y segundo semestre diecisiete (17)

Total de horas de la carrera Tecnicatura Universitaria en Programación Full Stack: 1696 (un mil seiscientas noventa y seis) horas reloj

b. Modalidad de dictado de las unidades curriculares

Todas las unidades curriculares son de modalidad presencial

c. Contenidos mínimos de las unidades curriculares

PRIMER AÑO

01. Introducción al Análisis Sistémico

Resolución de problemas básicos de lógica y razonamiento matemático. Representación de problemas matemáticos mediante ecuaciones y gráficos. Introducción a matrices y diagramas para modelar relaciones entre elementos. Definición de sistemas: características, componentes y tipos. Pensamiento sistémico: enfoque holístico frente al enfoque reduccionista. Modelos de sistemas: sistemas abiertos, cerrados, y cibernéticos. Aplicación del análisis sistémico en el desarrollo de software. Técnicas para identificar problemas en proyectos de desarrollo: observación directa. Análisis de requerimientos. Encuestas y entrevistas a stakeholders). Definición del problema: marco, alcance y restricciones. Herramientas para estructurar problemas (Árbol de problemas, Diagramas de Ishikawa -causa-efecto, Mapas mentales).

02. Bases de Datos 1

Introducción a las bases de datos y su importancia. Modelo Entidad-Relación (MER): entidades, relaciones, atributos, claves y cardinalidades. Creación de diagramas MER para diseño conceptual. Introducción a SQL: estructura del lenguaje, sintaxis básica y consultas simples. Comandos DDL (Data Definition Language): creación y modificación de tablas. Primeros pasos en SGBDR para implementar modelos básicos.

03. Emprendedurismo

Mercado laboral en el ámbito del Software. Modelo VICA y MR3 y su aplicación en el contexto profesional. Gestión personal de actividades y organización del tiempo. Formas de empleo, incluyendo trabajo remoto, freelance y en relación de dependencia. Emprendedurismo y desarrollo de proyectos propios. Conceptos básicos de contabilidad, economía y finanzas aplicados al sector tecnológico. Ética laboral y principios fundamentales para el ejercicio profesional. Conceptos básicos de negociación y resolución de conflictos. Propiedad intelectual, licencias de software y fundamentos del software libre. Funcionamiento y estructura organizacional de una empresa de software.

04. Alfabetización Digital

Conceptos fundamentales: ¿Qué es la alfabetización digital? Importancia en el ámbito profesional y técnico. Transformación digital y su impacto en el desarrollo de software. Tecnologías básicas: hardware, software, redes y sistemas operativos. Contexto del desarrollador Full Stack en el ecosistema digital. Introducción a entornos de desarrollo integrados (IDEs)

05. Algoritmos 1

Introducción a la lógica computacional y resolución de problemas. Estructuras básicas de datos y control de flujo. Diseño, implementación y análisis de algoritmos simples. Principios de recursividad. Algoritmos de búsqueda y ordenamiento básicos. Uso de estructuras iterativas y condicionales en la resolución de problemas. Representación gráfica de algoritmos mediante diagramas de flujo.

06. Arquitectura de Computadoras

Sistemas de numeración. Lógica binaria. Circuitos secuenciales y combinacionales. Organización de una computadora. Funcionamiento de la CPU. Código máquina. Tipos de arquitectura. Tipos de almacenamiento. Memoria. Buses internos y externos. Periféricos.

07. Bases de Datos 2

Modelo Relacional: transformación de MER a tablas relacionales. Profundización en SQL: consultas complejas, uso de funciones agregadas y operaciones con múltiples tablas (joins). Comandos DML (Data Manipulation Language): inserción, actualización y eliminación de datos. Introducción a vistas y restricciones de integridad. Uso de SGBDR para gestionar bases de datos relacionales con consultas y manipulación de datos.

08. Metodología de Resolución de Problemas

Fundamentos de resolución de problemas: Definición y naturaleza de un problema, tipos de problemas, diferencias entre síntomas y causas raíces. Observación, hipótesis, experimentación y análisis. Etapas y ciclo de resolución de problemas: Identificación, análisis, propuesta de soluciones, implementación y evaluación. Herramientas de apoyo: Análisis FODA. Matrices de decisión. Brainstorming y técnicas de ideación.

Técnicas y estrategias de resolución de problemas: Pensamiento crítico y lógico, identificación de sesgos y falacias, descomposición de problemas en partes manejables, técnicas específicas (5 porqués, método A3)

09. Programación 1

Introducción a la programación estructurada. Sintaxis y uso de un lenguaje de programación. Estructuras de control: condicionales, bucles y funciones. Estructuras de datos básicas: arreglos y cadenas. Uso de un entorno de desarrollo (IDE). Práctica en computadora con desarrollo, prueba y ejecución de programas simples.

10. Diseños y Arquitectura de Despliegue 1

Familiaridad con conceptos básicos de programación full stack (frontend y backend). Comprensión de fundamentos de redes, incluyendo protocolos como HTTP y DNS. Ciclo de vida del software: Desarrollo, pruebas, despliegue y mantenimiento. ¿Qué es una arquitectura de despliegue?. Introducción a los Tipos de despliegue: Local, en la nube, híbrido y Monolítico vs. arquitecturas distribuidas (microservicios). Introducción al Diseño de Infraestructura. 1) Modelos de despliegue: Servidor único vs. balanceadores de carga. Contenedores (Docker) y su rol en el despliegue. 2) Sistemas de virtualización: Introducción a máquinas virtuales y contenedores.

SEGUNDO AÑO

11. Diseños y Arquitectura de Despliegue 2

Introducción a los Patrones de diseño en despliegue (Alta disponibilidad, escalabilidad horizontal y vertical. Diseño para recuperación ante fallos). Integración Continua y Despliegue Continuo (CI/CD). Herramientas de automatización. Gestión de versiones y despliegues progresivos. Servicios en la Nube (Introducción a proveedores de servicios en la nube) Conceptos clave: Infraestructura como servicio (IaaS). Plataforma como servicio (PaaS). Seguridad en el despliegue: Configuración de certificados SSL/TLS. Gestión de credenciales y secretos (por ejemplo, en contenedores y entornos de nube). Monitorización: Herramientas para monitorear aplicaciones y servidores. Uso de logs para diagnóstico de problemas.

12. Programación Orientada a Objetos

Introducción al paradigma orientado a objetos: conceptos básicos y ventajas. Clases y objetos: definición, atributos y métodos. Encapsulación: acceso y control de visibilidad. Herencia: reutilización de código y jerarquías de clases. Polimorfismo: métodos

sobrescritos y sobrecarga. Abstracción: diseño de clases abstractas e interfaces. Uso de colecciones y estructuras dinámicas. Práctica de programación orientada a objetos en un lenguaje específico. Manejo de excepciones. Buenas prácticas en diseño orientado a objetos. Creación y análisis de proyectos básicos aplicando POO.

13. Base de Datos 3

Bases de Datos Relacionales avanzadas y No Relacionales. Optimización de consultas SQL: índices, planificación de consultas y subconsultas. Procedimientos almacenados, triggers y funciones en SQL. Introducción a bases NoSQL: diferencias con SQL y usos complementarios. Prácticas con bases de datos clave-valor, documentos y grafos. Integración de consultas SQL con sistemas híbridos relacional/no relacional.

14. Competencias Comunicacionales 1

Comprensión y producción de textos orales y escritos en contextos profesionales. Técnicas de indagación para la búsqueda de información y la interpretación de requerimientos. Medios y ruidos que afectan la comunicación, importancia de la retroalimentación. Oratoria y pitch: habilidades comunicacionales orales de acuerdo al contexto y destinatario. Dinámicas de grupos y equipos de trabajo: características, roles, estereotipos y rotación de roles. La tarea como eje de la convocatoria de equipos. Objetivos grupales, metas individuales, y la interacción entre lo individual y lo grupal. La empatía, la escucha activa y técnicas de resolución de conflictos. El equipo de proyectos de software: roles, responsabilidades y dinámicas específicas en el desarrollo de software. Elaboración de informes y presentación de resultados de manera clara y efectiva.

15. Programación Web

Introducción a la arquitectura cliente-servidor y protocolos HTTP/HTTPS. Diseño con HTML5: estructura, formularios y multimedia. Estilos con CSS3: diseño responsivo y maquetado adaptativo. JavaScript: interactividad, manipulación del DOM y manejo de asincronía con AJAX. Consumo de APIs REST y manipulación de datos JSON. Introducción a frameworks de JavaScript (React, Angular, Vue.js). Seguridad web: protección contra XSS, CSRF, y SQL Injection. Herramientas de desarrollo y despliegue de aplicaciones web.

16. Arquitectura de Internet y APIs REST

Fundamentos de la arquitectura de Internet: protocolos, servicios y modelos de comunicación. Protocolo HTTP/HTTPS: estructura, métodos y encabezados. Conceptos

de cliente y servidor, y su interacción en aplicaciones web. Introducción a APIs REST: definición, principios y diseño. Métodos RESTful y su uso en operaciones CRUD. Estructura de respuestas y manejo de códigos de estado HTTP. Autenticación y autorización en APIs REST. Prácticas de diseño para APIs escalables y seguras. Consumo de APIs desde aplicaciones cliente. Introducción a herramientas para prueba y documentación de APIs.

17.Programación Extrema y Desarrollo Guiado por Pruebas

Definiciones y principios fundamentales de Programación Extrema (XP). Prácticas clave de XP, como programación en pareja, propiedad colectiva del código, desarrollo iterativo y feedback continuo. Introducción al Desarrollo Guiado por Pruebas (TDD), sus conceptos y beneficios. Ciclo de TDD, incluyendo escritura de pruebas, implementación de código y refactorización. Uso de herramientas y entornos para TDD en el lenguaje de programación elegido. Diseño de casos de prueba efectivos y análisis de cobertura. Integración de TDD en metodologías ágiles. Análisis de casos prácticos y resolución de problemas mediante XP y TDD.

18.Sistemas Operativos

Tipos de Sistemas Operativos. Gestión de Procesos e Hilos. Planificación de Procesos. Gestión de Memoria. Paginación y Segmentación. Gestión de Almacenamiento. Sistemas de Archivos. Scripting en línea de comandos.

19.Competencias Comunicacionales 2

Relación entre comunicación y nuevas tecnologías: recorrido histórico y actualidad. Definición, características y desarrollo de las Nuevas Tecnologías de la Información y la Comunicación (NTIC). Influencia de las NTIC en lo individual, colectivo y profesional. Narrativas hipermedia y transmedia. Identidad digital profesional, redes profesionales y networking. Comunicación organizacional: definición, características y relación con cultura, aprendizaje y clima organizacional. Comunicación organizacional interna, externa, formal e informal. Comunicación en equipos de trabajo: roles, liderazgo, objetivos y resolución de conflictos. Participación inteligente y su impacto en el ámbito profesional. Debates, proyecciones y desafíos sobre el futuro de la comunicación.

20.Programación Backend

Definición y fundamentos del desarrollo backend. Introducción a lenguajes de programación y frameworks para backend. Arquitectura cliente-servidor y diseño de APIs. Manipulación de bases de datos: conexión, consultas y transacciones. Seguridad

en aplicaciones backend: autenticación, autorización y manejo de datos sensibles. Gestión de sesiones y estados. Optimización de rendimiento y escalabilidad. Práctica en implementación de soluciones backend integradas con servicios externos y aplicaciones frontend.

TERCER AÑO

21. Gestión de Proyectos

Concepto de proyecto y elementos clave de la gestión. Etapas de un proyecto y criterios para planificación. Métodos y herramientas de relevamiento y análisis de necesidades de usuarios. Planificación de proyectos: métodos PERT/CPM, diagramas de redes y de Gantt. Gestión de proyectos de software: estimaciones, herramientas y desarrollo del plan. Gestión de riesgos: tipos, impacto y planes de mitigación. Gestión de la calidad del software: conceptos, normas, validación y verificación. Gestión del mantenimiento: técnicas, herramientas y planificación. Gestión de configuraciones: etapas y software aplicado. Sustentabilidad en proyectos: eficiencia energética y uso racional de recursos naturales.

22. Desarrollo para Dispositivos Móviles y IoT

Definiciones y tecnologías para Frontend en dispositivos móviles. Lenguaje de programación aplicable. Diseño de UI con el framework elegido. ABP en el framework elegido.

23. Base de datos 4

SQL avanzado para sistemas distribuidos: replicación, partición y alta disponibilidad. Estrategias de tuning de bases de datos: índices avanzados, particionamiento y gestión de transacciones. Introducción a consultas analíticas en SQL (OLAP, CTE, ventanas). Gestión de datos masivos: Big Data y bases relacionales en arquitecturas distribuidas. Prácticas con sistemas como Hadoop y bases compatibles con SQL. Seguridad en SQL: encriptación, control de acceso y gestión de permisos. Análisis de casos complejos aplicando SQL y otros modelos de persistencia.

24. Habilidades para la Gestión

Oratoria: técnicas de presentación, voz y dicción, improvisación, lenguaje corporal, manejo de la audiencia, construcción de argumentos. Trabajo en equipo. Flexibilidad. Organización. Autonomía. Gestión de emociones. Resolución de problemas complejos. Pensamiento crítico. Inteligencia emocional.

25. Introducción a DevOps

Principios y fundamentos de DevOps. Cultura de colaboración entre desarrollo y operaciones. Introducción a la integración y entrega continua (CI/CD). Automatización de procesos en el ciclo de vida del software. Herramientas básicas para DevOps: gestión de versiones, pipelines y monitoreo. Práctica en entornos virtuales para implementación y despliegue inicial.

26. Verificación y Validación de Programas

Desarrollo guiado por pruebas (TDD). Pruebas Unitarias. Pruebas en navegadores Web. Pruebas de Integración. Pruebas de volumen. Pruebas de stress . Pruebas automatizadas. Documentación de los casos de prueba.

27. Estadística y Ciencia de Datos

Importancia de la estadística en la toma de decisiones. Medidas descriptivas (media, mediana, moda, rango, varianza). Distribuciones y correlaciones. Conceptos generales de ciencia de datos. Ciclo de vida del dato (adquisición, limpieza, análisis, visualización). Importancia de la ciencia de datos en aplicaciones full stack. Estructuras de datos: Tipos de datos comunes: Numéricos, categóricos, texto, y tiempo. Almacenamiento y acceso a datos en aplicaciones web. Integración de Ciencias de Datos en Aplicaciones Full Stack (Backend, Frontend y Automatización y pipelines de datos) Introducción a Machine Learning.

28. Práctica Profesionalizante 1

Diseño de un sistema como Trabajo Final. Programación y pruebas del backend. Proyectos de investigación aplicados a la práctica profesional del desarrollo full stack. Abordaje de problemas específicos en contextos reales aplicando los conocimientos teóricos y metodológicos. Identificación de las variables del entorno organizacional. Generación de propuestas de valor significativas, reales, medibles y aplicables a las demandas de las organizaciones locales y regionales. Diseño de presupuestos profesionales y gestión de cobros. Enfoque práctico. Trabajo en equipo. Manejo de tiempos y de trabajo interdisciplinario con carreras afines. Reconocimiento de nuevas tendencias de mercado y diseños de trabajo aplicados a realidades particulares. Programación y pruebas del frontend. Documentación del Trabajo Final.

29. Ética y Deontología

Responsabilidad social en el ejercicio profesional. Normas y valores morales que los profesionales de un determinado sector deben respetar durante el ejercicio de su

profesión. Introducción a la ética profesional. Principios éticos fundamentales. Códigos de ética: concepto, ejemplos según las distintas profesiones. Aplicación. Dilemas éticos comunes. Ética y responsabilidad social. Confidencialidad y privacidad. Ética en la toma de decisiones. Impacto de la tecnología en la Ética profesional. Casos de estudios y análisis. Conclusiones y Buenas Prácticas. Consideraciones éticas en la creación de contenido digital. Protección de derechos de autor y propiedad intelectual. Prácticas responsables en la comunicación digital. Ley 25326. Conceptos básicos de Ciberseguridad. Internet y las redes sociales

3.4. Propuesta de seguimiento curricular

El/la responsable académico/a de la carrera estará a cargo de la organización y gestión de la tecnicatura, con el fin de alcanzar los objetivos y el perfil profesional propuesto. Asimismo, será responsable del seguimiento e implementación del plan de estudios y de su revisión periódica. Tendrá injerencia en acciones de gestión académica como la conformación de equipos, cumplimiento de los programas de las asignaturas, seguimiento de la formación teórica y práctica brindada a los estudiantes, métodos de enseñanza y formas de evaluación, entre otros aspectos.



Esp. María Julia Oliva Cúneo
Rectora Normalizadora
Universidad Provincial de Córdoba

A handwritten signature in black ink, which appears to be 'Julia', is written over the typed name and title of the Rectora Normalizadora.