

SALTA, 11-MAR-2024

RESOLUCIÓN Nº 111

UNIVERSIDAD PROVINCIAL DE LA ADMINISTRACIÓN, TECNOLOGÍA Y OFICIOS

Expediente SICAD Nº 75/24

VISTO el Artículo 75, apartado 19, de la Constitución Nacional, la Ley nacional 24521 y la Ley 8312, modificatoria de su similar 7803, y

CONSIDERANDO:

Que, de conformidad con lo dispuesto por el Artículo 26 de la Ley 24521, la enseñanza superior universitaria estará a cargo de las Universidades Nacionales, de las Universidades Provinciales y Privadas, y de los Institutos Universitarios;

Que, mediante Ley 8312, modificatoria de su similar 7803, se crea la Universidad Provincial de la Administración, Tecnología y Oficios (UPATecO) como persona jurídica pública, con autonomía institucional y académica, y autarquía financiera y administrativa;

Que, por su parte, el Artículo 27 de la Ley 7803, modificado por Ley 8312, establece que los títulos que emita la UPATecO tendrán validez en todo el territorio de la provincia de Salta y habilitarán a los egresados a ejercer su profesión u oficio, conforme la normativa vigente en la materia;

Que, en las actuaciones de referencia, la Dirección General de Tecnología y Oficios de esta Universidad, eleva el Proyecto de Plan de Estudios correspondiente a la carrera de pregrado "Tecnatura Universitaria en Programación con Orientación en Desarrollo de Aplicaciones";

Que dicha carrera se organiza en base al Modelo de Formación Modular y Certificación por Competencias, con módulos que conforman unidades académicas mínimas;

Que la carrera mencionada posibilitará que sus egresados adquieran habilidades, aptitudes y competencias relacionadas a la creación, diseño, programación y desarrollo de aplicaciones móviles, web o de escritorio, innovadoras y funcionales, participando en las distintas fases del ciclo de vida del software, para brindar respuesta a las demandas evidenciadas tanto a nivel local como global;

Que en autos queda claramente establecida la fundamentación de la necesidad de la carrera, duración, perfil del egresado, alcance laboral, objetivos de la carrera, evaluación, organización curricular, contenidos mínimos de cada espacio curricular, entre otros ítems;

.. //

.. // RESOLUCIÓN N° 111

Expediente SICAD N° 75/24

Que la Dirección Jurídica de esta Universidad se expide favorablemente a través de Dictamen N° 69/24, por lo que corresponde el dictado del acto administrativo pertinente, en virtud de la Ley 8312, modificatoria de su similar 7803, y del Decreto N° 88/23;

Por ello;

**EL RECTOR DE LA UNIVERSIDAD PROVINCIAL
DE LA ADMINISTRACIÓN, TECNOLOGÍA Y OFICIOS**

R E S U E L V E :

ARTÍCULO 1º.- Aprobar el Plan de Estudios de la carrera de pregrado “Tecnatura Universitaria en Programación con Orientación en Desarrollo de Aplicaciones”, de la Universidad Provincial de la Administración, Tecnología y Oficios (UPATecO), que como Anexo forma parte de este instrumento legal.

ARTÍCULO 2º.- La presente resolución será refrendada por la señora Vicerrectora de esta Universidad.

ARTÍCULO 3º.- Comunicar y archivar.

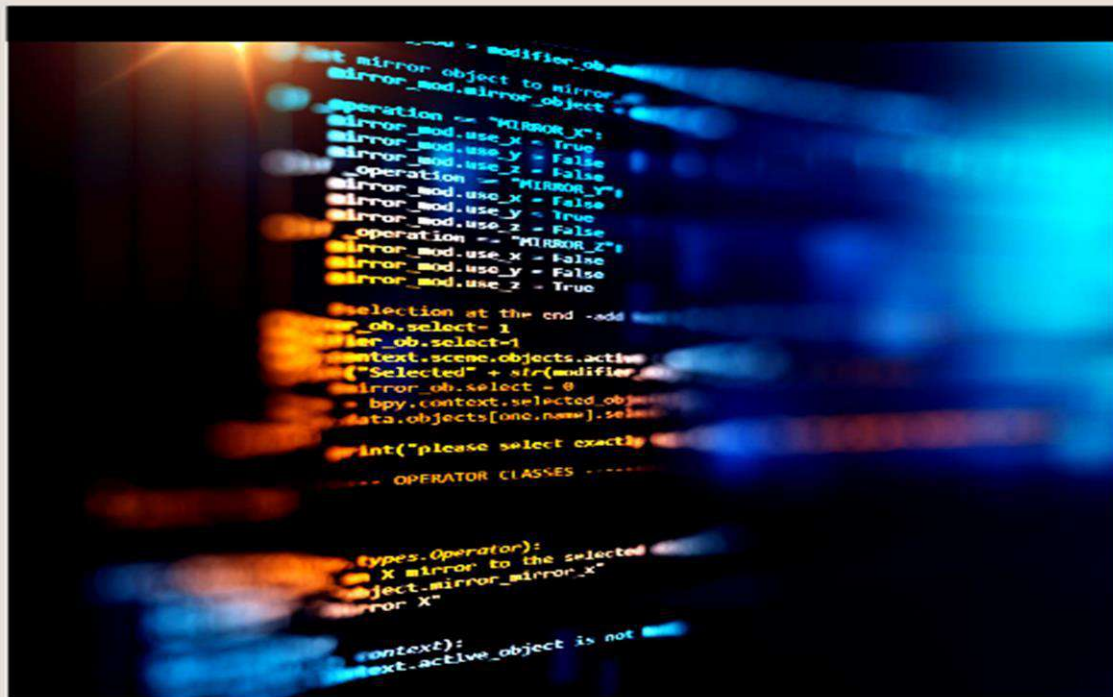


Firmado digitalmente por
Dra. MARÍA de los D. TALENS
Vicerrectora UPATecO



Firmado digitalmente por
Dr. CARLOS MORELLO
Rector UPATecO

RC
DD



*Tecnicatura
universitaria en
Programación
con orientación
en Desarrollo de
Aplicaciones*

**UNIVERSIDAD PROVINCIAL DE
ADMINISTRACIÓN, TECNOLOGÍA Y OFICIOS**

AUTORIDADES



RECTOR
Dr. Carlos Morello

VICERRECTORA

Dra. María de los D. Talens



PLAN DE ESTUDIOS 2024

IDENTIFICACIÓN DE LA CARRERA

Institución	Universidad Provincial de Administración, Tecnología y Oficios (UPATecO)
Unidad Académica	Unidad de Tecnología y Oficios

1.- CARACTERÍSTICAS DE LA CARRERA

Denominación de la Carrera

Tecnicatura Universitaria en Programación con orientación en desarrollo de Aplicaciones

Nivel Académico

Tecnicatura Superior Universitaria

Modalidad

Híbrida

Localización de la propuesta

Barrio Pereyra Rozas Calle s/n – Salta Capital.

Duración de la Carrera

2 años y medio.

Régimen de Cursado

Cuatrimestral

Título que Otorga

Técnico Universitario en Programación con orientación en desarrollo de Aplicaciones

Carga Horaria Total

1792 horas reloj

2.- FUNDAMENTACION DE LA CARRERA

La Tecnicatura Universitaria en Programación con orientación en desarrollo de Aplicaciones emerge como respuesta a la creciente demanda en la industria tecnológica, donde la innovación y la digitalización son imperativos.

Su origen se remonta al reconocimiento de la necesidad de formar profesionales capaces de concebir, diseñar y programar aplicaciones Informáticas que impulsen el progreso en diversos ámbitos.

En un mundo cada vez más interconectado, la relevancia de esta carrera es innegable. Las aplicaciones móviles, plataformas web y software especializado, son pilares fundamentales en campos tan diversos como la educación, la salud, el comercio electrónico y la gestión empresarial.

En las últimas décadas, el desarrollo de aplicaciones ha experimentado un crecimiento exponencial que ha transformado profundamente la forma en que interactuamos con la tecnología y el mundo que nos rodea. Este fenómeno ha sido impulsado por una serie de factores claves que han convergido para crear un entorno propicio para la innovación y la expansión del sector de desarrollo de software.

En primer lugar, el avance acelerado de la tecnología ha permitido la proliferación de dispositivos móviles, la conectividad ubicua a Internet y el acceso a potentes plataformas de computación en la nube. Estos avances han democratizado el desarrollo de aplicaciones, permitiendo que prácticamente cualquier persona con conocimientos básicos de programación pueda crear y distribuir software a una escala global.

Además, la creciente demanda de soluciones digitales por parte de empresas, organizaciones y consumidores ha impulsado la necesidad de aplicaciones innovadoras que satisfagan una amplia gama de necesidades y requerimientos. Desde aplicaciones de productividad y entretenimiento hasta soluciones empresariales y de atención médica, el desarrollo de aplicaciones ha diversificado su alcance y su impacto en la sociedad.

La proliferación de plataformas y herramientas de desarrollo, junto con la adopción generalizada de metodologías ágiles y prácticas de desarrollo iterativo, ha facilitado el proceso de creación y lanzamiento de aplicaciones al mercado. Esto ha permitido a los desarrolladores adaptarse rápidamente a los

cambios en los requisitos del usuario y en el panorama tecnológico, acelerando el ritmo de la innovación y la mejora continua.

Además, el desarrollo de aplicaciones ha generado nuevas oportunidades económicas y laborales en todo el mundo, impulsando la creación de startups tecnológicas, incubadoras de innovación y ecosistemas emprendedores. La industria del desarrollo de software se ha convertido en un motor de crecimiento económico y una fuente de empleo para profesionales de la tecnología en campos como la programación, el diseño de interfaces de usuario, la gestión de proyectos y la seguridad informática.

La Tecnicatura Universitaria en Programación con orientación en Desarrollo de Aplicaciones prepara a los estudiantes para abordar los desafíos de esta era digital, dotándolos de los conocimientos y habilidades necesarios para enfrentarse a un mercado laboral en constante evolución.

La importancia de esta carrera radica en su capacidad para impulsar la innovación y la transformación digital. Los graduados no solo tienen la capacidad de desarrollar soluciones tecnológicas, sino también de adaptarse y anticiparse a las nuevas tendencias y tecnologías emergentes. Su trabajo no se limita a la creación de aplicaciones, sino que contribuye activamente al desarrollo económico, social y tecnológico de la sociedad en su conjunto.

La Tecnicatura Universitaria en Programación con orientación en desarrollo de Aplicaciones es mucho más que una simple formación técnica; es un motor de cambio y progreso en la era digital.

3.- PERFIL DEL EGRESADO

El perfil profesional destaca las habilidades, conocimientos y competencias adquiridas durante el proceso de formación académica. Este perfil describe las capacidades específicas que el egresado ha desarrollado a lo largo de sus estudios y que lo hacen apto para desempeñarse en el campo laboral del desarrollo de aplicaciones.

El egresado consolidará una combinación de habilidades técnicas, creativas y analíticas que son relevantes para el diseño, la creación y la implementación de aplicaciones informáticas.

Entre los elementos claves que formarán parte de este perfil debemos

mencionar el dominio de lenguajes de programación y manejo de frameworks y herramientas de desarrollo de software; el diseño de interfaces de usuario (UI) y experiencia de usuario (UX), resaltando la capacidad para crear interfaces atractivas, funcionales e intuitivas con conocimientos en diseño gráfico, usabilidad y accesibilidad. Además, se hará un fuerte hincapié en el desarrollo de aplicaciones móviles, fortaleciendo la experiencia en la creación de aplicaciones para dispositivos móviles, incluyendo el desarrollo front-end y back-end. Se consolidará cierta familiaridad con metodologías ágiles de desarrollo de software (Scrum), control de versiones (Git) y bases de datos.

El egresado tendrá una visión integral, capacidad para identificar y solucionar problemas técnicos en el desarrollo de aplicaciones, así como para realizar pruebas y depuraciones eficientes; adaptabilidad y aprendizaje continuo, sobre todo teniendo en cuenta que la industria tecnológica evoluciona rápidamente, por lo que la disposición para aprender nuevas tecnologías y adaptarse a los cambios es fundamental.

Durante sus prácticas se enfrentarán a proyectos relevantes en donde podrán demostrar habilidades y creatividad en el diseño y desarrollo de aplicaciones y el trabajo en equipos multidisciplinarios, logrando una comunicación efectiva y gestión de proyectos.

El perfil del graduado en esta Tecnicatura Universitaria en Programación con orientación en desarrollo de Aplicaciones le permite analizar y diseñar futuras aplicaciones y por otro lado participar en el desarrollo de estas empleando metodologías y tecnologías del ámbito específico.

Es dable destacar que entre las competencias que se requieren para el diseño y desarrollo de aplicaciones que deben desarrollar los estudiantes se incluyen:

- Programación: capacidad para escribir código en varios lenguajes de programación relevantes para el desarrollo de aplicaciones, como JavaScript, Python, Java, C#, entre otros.
- Diseño de interfaces de usuario (UI) y experiencia de usuario (UX): habilidad para crear interfaces atractivas, funcionales e intuitivas, centrándose en la experiencia del usuario final.

- Desarrollo de aplicaciones móviles y web: competencias para desarrollar aplicaciones tanto para dispositivos móviles como para plataformas web, utilizando tecnologías actuales y frameworks relevantes.
- Gestión de bases de datos: conocimientos sobre diseño, administración y manejo de bases de datos, así como lenguajes de consulta como SQL para interactuar con ellas.
- Metodologías ágiles: familiaridad con metodologías de desarrollo ágil como Scrum para gestionar proyectos y equipos de manera efectiva.
- Resolución de problemas: habilidad para identificar, analizar y solucionar problemas técnicos durante el ciclo de vida del desarrollo de aplicaciones.
- Trabajo en equipo: capacidad para colaborar y comunicarse eficazmente dentro de equipos multidisciplinarios, contribuyendo al éxito de los proyectos.
- Adaptabilidad y aprendizaje continuo: disposición para aprender nuevas tecnologías y conceptos en un entorno tecnológico en constante evolución.
- Prácticas éticas y de seguridad: conciencia sobre la importancia de prácticas éticas en el desarrollo de aplicaciones y comprensión de las medidas de seguridad necesarias para proteger los sistemas y datos.

Estas competencias preparan al graduado para ingresar al mercado laboral en roles relacionados con el desarrollo de aplicaciones, ya sea como desarrollador de software, diseñador de interfaces, analista de sistemas o en otros roles afines dentro de la industria tecnológica.

El enfoque de "aprender haciendo" es ampliamente recomendado en este caso dado que ofrece una serie de beneficios significativos para los estudiantes tales como la aplicación de los conceptos teóricos aprendidos en el aula virtual a situaciones y problemas reales, desarrollar habilidades

prácticas esenciales, como programación, diseño de interfaces, resolución de problemas, trabajo en equipo y gestión de proyectos. La participación en proyectos y actividades prácticas suele conducir a una comprensión más profunda y duradera de los conceptos, en comparación con el aprendizaje pasivo basado solo en la teoría.

Es fundamental la preparación para el mundo laboral, al adquirir experiencia práctica durante sus estudios, los estudiantes están mejor equipados para enfrentar desafíos en el entorno laboral y para adaptarse rápidamente a las demandas del mercado. Además de habilidades técnicas, el enfoque de "aprender haciendo" fomenta habilidades como la resolución de problemas, la creatividad, la comunicación y el trabajo en equipo, habilidades fundamentales en el ámbito laboral. La participación en proyectos prácticos puede aumentar la motivación y el compromiso de los estudiantes, ya que ven la relevancia y aplicación directa de lo que están aprendiendo.

El enfoque de "aprender haciendo" es altamente beneficioso para estudiantes de desarrollo de aplicaciones, ya que les proporciona una base sólida de habilidades prácticas y experiencia que son valiosas en la industria tecnológica. Este enfoque les permite estar mejor preparados para enfrentar los desafíos del mercado laboral una vez que se gradúan.

Es esencial que el egresado demuestre su capacidad para adaptarse a las demandas cambiantes del sector tecnológico y para seguir aprendiendo y mejorando sus habilidades a lo largo de su carrera profesional.

4.- ALCANCE DEL PERFIL LABORAL

El Técnico Universitario en Programación con orientación en Desarrollo de Aplicaciones está preparado para contribuir en el desarrollo, implementación y mantenimiento de aplicaciones informáticas, así como para trabajar en equipo en proyectos tecnológicos en diferentes entornos laborales, como empresas de software, agencias digitales, startups o como profesional independiente.

El trabajo de un desarrollador de aplicaciones abarca varias etapas desde la concepción de la idea hasta la creación y lanzamiento del producto final y sus futuras mejoras. Aquí se detallan las etapas típicas involucradas en este proceso:

- Investigación y conceptualización:

Idea inicial: Comienza con la generación de la idea para la aplicación, identificando una necesidad específica del mercado o una solución a un problema.

Investigación de mercado y usuarios: Realización de investigaciones para comprender el mercado objetivo, analizar la competencia y conocer las necesidades y preferencias de los usuarios.

Definición de objetivos y funcionalidades: Establecimiento de los objetivos de la aplicación y la definición de las características y funcionalidades principales que se requerirán.

- Diseño y planificación:

Diseño de la experiencia de usuario (UX) y la interfaz de usuario (UI): Creación de prototipos y diseños visuales que representen la estructura y apariencia de la aplicación, centrándose en la facilidad de uso y la estética.

Arquitectura de la aplicación: Definición de la estructura técnica y la arquitectura de la aplicación, incluyendo la selección de tecnologías, bases de datos, servidores, etc.

Planificación del desarrollo: Establecimiento de los plazos, recursos necesarios y asignación de tareas para el equipo de desarrollo.

- Desarrollo:

Codificación y programación: Escritura del código fuente de la aplicación basándose en los diseños y especificaciones previamente definidos.

Pruebas y depuración: Realización de pruebas exhaustivas para identificar y corregir errores (bugs) y asegurar que la aplicación funcione correctamente en diferentes dispositivos y escenarios.

Iteraciones y mejoras: Se realizan ajustes y mejoras basados en la

retroalimentación obtenida durante las pruebas.

- Lanzamiento y post-lanzamiento:

Despliegue: Publicación de la aplicación en las tiendas de aplicaciones (App Store, Google Play, etc.) o en la web.

Marketing y promoción: Creación de estrategias para promocionar la aplicación y aumentar su visibilidad entre los usuarios.

Recopilación de comentarios: Monitoreo de la retroalimentación de los usuarios después del lanzamiento para identificar áreas de mejora y posibles actualizaciones.

- Mantenimiento y actualización:

Parches y actualizaciones: Implementación de parches de seguridad, corrección de errores y lanzamiento de actualizaciones para mejorar la aplicación y agregar nuevas funcionalidades.

El desarrollador de aplicaciones trabaja en equipo, colaborando estrechamente con otros profesionales como diseñadores gráficos, ingenieros de software, especialistas en experiencia de usuario (UX) y especialistas en marketing, entre otros, para llevar a cabo cada etapa del proceso y lograr un producto final exitoso que cumpla con las necesidades de los usuarios y el mercado.

5.- ÁREA OCUPACIONAL

El área ocupacional de un programador con orientación en desarrollo de aplicaciones abarca una amplia gama de roles, responsabilidades y habilidades que se centran en el desarrollo y mantenimiento de aplicaciones informáticas. Estos profesionales trabajan en la creación de aplicaciones móviles, web o de escritorio, participando en diferentes fases del ciclo de vida del software.

Entre las principales áreas ocupacionales se encuentran:

Diseño de Experiencia de Usuario y Diseño de Interfaz de Usuario:

UX Designer (Diseñador de Experiencia de Usuario): Se enfoca en comprender las necesidades del usuario y crea experiencias de usuario intuitivas, centradas en la usabilidad y la interacción del usuario con la aplicación.

UI Designer (Diseñador de Interfaz de Usuario): Diseña la apariencia visual de la aplicación, creando interfaces atractivas y funcionales que se alinean con la identidad de la marca y los principios de diseño.

Desarrollo de Software:

Desarrollador Front-end: Crea la parte visible de la aplicación que interactúa directamente con el usuario, utilizando lenguajes de programación como HTML, CSS y JavaScript para implementar diseños y funcionalidades.

Desarrollador Back-end: Se encarga de la lógica y la funcionalidad detrás de la interfaz de usuario, trabajando con bases de datos, servidores y aplicando lenguajes como Python, Java, PHP, entre otros.

Gestión de Proyectos y Coordinación de Equipos:

Project Manager (Gestor de Proyectos): Supervisa el desarrollo de la aplicación, gestiona plazos, recursos y coordina al equipo para alcanzar los objetivos establecidos.

Scrum Master: Facilita el uso de metodologías ágiles, como Scrum, dentro del equipo de desarrollo, asegurando una comunicación efectiva y la entrega continua de funcionalidades.

Calidad y Pruebas:

Tester/QA (Control de Calidad): Realiza pruebas exhaustivas para identificar errores, problemas de rendimiento y funcionalidades defectuosas antes del lanzamiento de la aplicación.

Seguridad Informática y Mantenimiento:

Security Analyst (Analista de Seguridad): Se encarga de identificar y mitigar vulnerabilidades de seguridad en la aplicación para protegerla contra amenazas y ataques cibernéticos.

DevOps Engineer: Se centra en la automatización de procesos, el despliegue continuo y la gestión de la infraestructura de la aplicación para garantizar su disponibilidad y funcionamiento óptimo.

Estas áreas ocupacionales pueden solaparse en ciertas circunstancias y los profesionales pueden tener habilidades multidisciplinarias que les permiten trabajar en más de un rol dentro del desarrollo de aplicaciones.

En síntesis, el área ocupacional de un desarrollador de aplicaciones involucra diversas habilidades técnicas, creativas, de gestión y de coordinación para llevar a cabo con éxito el desarrollo de aplicaciones informáticas innovadoras y funcionales.

6.- OBJETIVOS DE LA CARRERA

Objetivo General

La UPATecO tiene por objetivo la formación tecnológica de hombres y mujeres de conformidad a nuevas demandas y necesidades formativas de habilidades y/o competencias solicitadas por los sectores productivos de la Provincia. Por tanto, la presente Tecnicatura promueve el desarrollo y la promoción de la economía del conocimiento como motor de desarrollo económico en la provincia de Salta, capacitando recursos humanos de alta capacidad en el área del desarrollo de aplicaciones como respuesta a una necesidad puntual del sector, respondiendo así a las demandas evidenciadas a nivel tanto local como global.

Objetivos Específicos

En concordancia con el objetivo general, se plantean los siguientes objetivos específicos:

- Formar técnicos universitarios en desarrollo de aplicaciones competentes para crear, analizar, diseñar, desarrollar, implementar y probar aplicaciones informáticas destinadas a distintos tipos de organizaciones.
- Capacitar técnicos competentes que puedan desarrollar aplicaciones que se ajusten a las necesidades de organizaciones de distinta índole.
- Formar técnicos universitarios capaces de insertarse en las distintas fases del desarrollo de aplicaciones, desde la creación y diseño, hasta el desarrollo, pasando por la programación, el despliegue y la comercialización.
- Capacitar a estos profesionales en la utilización de herramientas tecnológicas adecuadas: lenguajes de programación, bases de datos, metodologías ágiles, entre otras.
- Formar técnicos universitarios capaces de utilizar las metodologías actuales para el trabajo colaborativo: metodologías ágiles, etc.

7.- REQUISITOS DE INGRESO

Podrán cursar esta tecnicatura universitaria aquellas personas que:

- Posean título secundario o equivalente completo, cualquiera sea su modalidad, emitidos por instituciones de gestión estatal o privada y consten con el debido reconocimiento ministerial, conforme lo establece el artículo 7º de la Ley de Educación Superior N° 24521.
- Quienes no posean título secundario o equivalente, pero que sean mayores de 25 años y se encuentren en el marco de excepcionalidad establecido en la segunda parte del artículo 7º de la Ley de Educación Superior N° 24521.
- Acrediten título secundario completo, emitido por otro país, pero debidamente reconocido por el Ministerio de Relaciones Exteriores y Culto y el Ministerio de Educación de la Nación.
- Registren su preinscripción en las fechas establecidas según resolución rectoral.

- Dispongan de una PC o equipo informático con una conexión a Internet estable para acceder a la plataforma educativa, las clases, los contenidos, los trabajos prácticos y demás actividades de la cursada, como también a los exámenes finales.

8.- Evaluación

De acuerdo al modelo formativo adoptado por la Universidad Provincial de Administración Pública, Tecnología y Oficios las unidades académicas ofrecen su definición, objetivos de aprendizaje y evidencias de dominio de los contenidos mínimos, en estructuras denominadas normas de competencia, que funcionan como un instrumento para la evaluación y certificación de competencias del estudiante. Cada evidencia de competencia debe tratarse de una única acción, decisión, proceso o actitud que pueda evidenciar y observar el formador en cada estudiante.

Los estudiantes estarán en condiciones de acceder a la evaluación final en condición de Certificables del Módulo si reúnen los siguientes requisitos:

- 80% asistencia obligatoria a clases sincrónica y virtual ó el 70% cuando sean justificadas las inasistencias por cuestiones de salud u otras razones de fuerza mayor.
- 1 examen parcial aprobado, en caso de los módulos cuatrimestrales. Con esta condición quedarían Certificables para la materia.
- Para Certificar el Módulo se requiere la aprobación de su correspondiente Laboratorio, en donde los estudiantes deberán dar cuenta de las competencias adquiridas en su desarrollo.

Todos los módulos deben tener una instancia de parciales para acceder al Laboratorio de Certificación.

En el caso de estudiantes libres que quieran acreditar un módulo, dicho Laboratorio deberá constar de una instancia escrita en la cual se evaluará la parte práctica y luego una parte oral, para la parte teórica. Se debe aprobar indefectiblemente ambas instancias para acreditar el Módulo.

Los talleres, seminarios y prácticas profesionales pueden ser Certificable-Certificado, según el criterio del Formador y las características del Módulo. Siendo el mismo el responsable de establecer las exigencias académicas correspondientes.

9.- ORGANIZACIÓN CURRICULAR

Plan de Estudios con carga horaria y correlatividades

PRIMER AÑO (1° Cuatrimestre)								
Código	Módulos	Régimen	Horas Semanales por Cuatrimestre		Total Horas Cuatrimestrales	Correlativas		
			1°	2°		para Cursar		Aprobadas para Rendir
						Regulares	Aprobadas	
101	Fundamentos de Programación I	Cuat	8	-	128	---	---	---
102	Inglés técnico I	Cuat	4	-	64	---	---	---
103	Matemática computacional	Cuat	3	-	48	---	---	---
104	Taller de Informática	Cuat	3	-	48	---	---	---
105	Tecnologías de la Información en las organizaciones	Cuat	6	-	96	---	---	---
PRIMER AÑO (2° Cuatrimestre)								
106	Fundamentos de Programación II	Cuat	-	8	128	---	---	101
107	Análisis y Diseño de Sistemas I	Cuat	-	6	96	---	---	101,105
108	Base de Datos I	Cuat	-	4	64	---	---	106,103
109	Seminario Tecnológico I	Cuat	-	3	48	---	---	104
110	Práctica Profesional I	Cuat	-	3	48	---	---	105,106,107,108

SEGUNDO AÑO (1° Cuatrimestre)								
Código	Módulos	Régimen	Horas Semanales por Cuatrimestre		Total Horas Cuatrimestrales	Correlativas		
			1°	2°		para Cursar		Aprobadas para Rendir
						Regulares	Aprobadas	
2001	Programación Móvil I	Cuat	4	-	64	---	---	106
2002	Inglés técnico II	Cuat	4	-	64	---	---	102
2003	Programación Web I	Cuat	4	-	64	---	---	106
2004	Base de datos II	Cuat	4	-	64	---	---	108
2005	Marketing de Aplicaciones I	Cuat	6	-	96	---	---	105,109
SEGUNDO AÑO (2° Cuatrimestre)								
2006	Programación Móvil II	Cuat	-	4	64	---	---	201
2007	Análisis y Diseño de Sistemas II	Cuat	-	6	80	---	---	107,205
2008	Programación Web II	Cuat	-	4	64	---	---	203
2009	Seminario Tecnológico II	Cuat	-	3	48	---	---	109

9								
210	Práctica Profesional 2	Cuat	-	3	48	---	---	110,206,207,208

TERCER AÑO (1° Cuatrimestre)								
Código	Módulos	Régimen	Horas Semanales por Cuatrimestre		Total Horas Cuatrimestrales	Correlativas		
			1º	2º		para cursar		Aprobadas para Rendir
						Regulares	Aprobadas	
301	Marketing de Aplicaciones II	Cuat	6	-	48	---	---	205,20 7,210
302	Inglés técnico III	Cuat	4	-	64	---	---	202
303	Ética y legislación laboral	Cuat	3	-	48	---	---	301
304	Seminario Tecnológico III	Cuat	3	-	48	---	---	209
305	Práctica Profesional	Cuat	3	-	48	---	---	210,30 3



10.- CONTENIDOS MÍNIMOS

1. 101. Fundamentos de Programación I

Unidad 1: Introducción a la programación y Python. Breve historia de la programación. Qué es Python y por qué es popular. Instalación de Python y entornos de desarrollo. Primer programa en Python: ¡Hola Mundo!

Variables y tipos de datos. Operaciones básicas con variables y tipos de datos. Estructuras de control de flujo: Condiciones: if, elif, else. Bucles: while, for. Uso de break y continue.

Unidad 2: Listas y tuplas. Creación y manipulación de listas y tuplas. Indexación y slicing. Métodos y funciones comunes. Diccionarios y conjuntos. Creación y manipulación. Acceso a elementos y métodos útiles.

Unidad 3: Funciones y modularidad: Definición y llamada de funciones. Paso de argumentos a funciones. Retorno de valores desde funciones. Modularidad y organización de código en funciones.

Manejo de errores y excepciones. Uso de try, except y finally. Manejo de excepciones personalizadas.

Unidad 4: Introducción a la programación orientada a objetos (POO):

Conceptos básicos de POO. Clases y objetos en Python. Atributos y métodos de clase. Herencia y polimorfismo básico.

Unidad 5: Manipulación de archivos. Abrir, leer y escribir archivos en Python. Manipulación de datos estructurados en archivos (JSON, CSV).

Introducción a bibliotecas y módulos en Python. Importación de módulos.

Uso de bibliotecas estándar de Python (math, random). Instalación y uso de bibliotecas externas a través de pip.

2. 102. Inglés Técnico I

Unidad 1: Importancia del inglés en la carrera de Programación. Introducción al vocabulario básico relacionado con la tecnología y el desarrollo de software.

Unidad 2: Fundamentos de gramática y estructuras de oraciones. Presente simple y presente continuo: uso y estructura. Pasado simple y pasado continuo: uso y estructura. Estructuras de preguntas y respuestas en presente y pasado. Uso de los pronombres personales y posesivos.

Unidad 3: Comprensión de lectura y escritura de documentación técnica. Redacción de mensajes de correo electrónico y documentación técnica básica. Comprensión de instrucciones y tutoriales en inglés para resolver problemas de programación.

Unidad 4: Desarrollo de habilidades de escucha. Escucha de conferencias y charlas en inglés sobre tecnología y desarrollo de software. Comprensión de podcasts y entrevistas con profesionales de la industria. Realización de ejercicios de escucha para mejorar la comprensión auditiva y el reconocimiento de vocabulario técnico.

3. 103. Matemática computacional

Unidad 1: Introducción a la lógica y su relación con la programación. Importancia en la resolución de problemas. Conceptos básicos de lógica proposicional. Proposiciones y conectivas lógicas. Tablas de verdad y tautologías. Leyes y reglas de inferencia.

Unidad 2: Lógica de predicados. Introducción a los predicados y cuantificadores (existencial y universal). Cuantificación y sustitución. Inferencia y demostración en lógica de predicados.

Unidad 3: Razonamiento lógico en programación. Uso de la lógica en la toma de decisiones en programación. Condicionales y bucles como expresiones lógicas. Construcción de expresiones lógicas en programación.

Algoritmos y resolución de problemas lógicos. Desarrollo de algoritmos para resolver problemas lógicos simples. Estrategias de resolución de problemas lógicos. Implementación de algoritmos en un lenguaje de programación.

Unidad 4: Lógica booleana y álgebra de Boole. Definición de álgebra de Boole y sus operaciones básicas. Simplificación de expresiones booleanas. Aplicaciones en diseño de circuitos lógicos y programación.

Unidad 5: Diagramas de Venn y representaciones gráficas de lógica. Representación gráfica de proposiciones y relaciones lógicas. Lógica modal y su aplicación en la programación. Operadores modales (necesidad, posibilidad, conocimiento, creencia).

4. 104. Taller de Informática

Unidad 1: Introducción a la informática y herramientas de ofimáticas. Importancia en el ámbito laboral. Conceptos básicos de ofimática: definición, historia y evolución.

Herramientas de oficina más comunes: procesadores de texto, hojas de cálculo, programas de presentaciones.

Unidad 2: Procesadores de texto. Introducción a los procesadores de texto: Microsoft Word, Google Docs, LibreOffice Writer. Funciones básicas. Formato de texto. Herramientas de edición de texto. Funciones avanzadas: tablas, imágenes, encabezados y pies de página, notas al pie.

Unidad 3: Hojas de cálculo. Introducción a las hojas de cálculo: Microsoft Excel, Google Sheets, LibreOffice Calc. Estructura de una hoja de cálculo. Ingreso de datos y formatos numéricos. Funciones básicas. Fórmulas y funciones avanzadas. Gráficos: creación y personalización de gráficos de barras, líneas, pasteles.

Unidad 4: Programas de presentaciones. Introducción a los programas de presentaciones: Microsoft PowerPoint, Google Slides, LibreOffice Impress. Creación de diapositivas. Diseño de diapositivas: fondos, temas, disposición. Animaciones y transiciones. Presentación de diapositivas.

Unidad 5: Organización y gestión de archivos. Organización de archivos y

carpetas en el sistema operativo. Uso de herramientas de gestión de archivos. Comprimir y descomprimir archivos: formatos ZIP, RAR. Gestión de versiones y control de cambios en documentos.

Unidad 6: Colaboración y trabajo en equipo. Compartir documentos en la nube: Google Drive, Microsoft OneDrive. Colaboración en tiempo real: edición simultánea de documentos. Comentarios y revisiones: revisión de cambios, aceptación o rechazo de modificaciones.

Uso de herramientas de comunicación: correo electrónico, mensajería instantánea, videoconferencia.

5. 105. Tecnologías de la Información en las organizaciones

Unidad 1: Introducción a las Tecnologías de la Información (TI) en las Organizaciones. Concepto de Tecnologías de la Información (TI). Importancia de las TI en las organizaciones. Evolución histórica de las TI en el ámbito empresarial. Impacto de las TI en los procesos empresariales y en la toma de decisiones.

Unidad 2: Infraestructura tecnológica en las Organizaciones. Componentes de la infraestructura tecnológica: hardware, software, redes, almacenamiento. Arquitectura de sistemas de información. Gestión de la infraestructura tecnológica: adquisición, mantenimiento, seguridad. Tendencias en infraestructura tecnológica: nube, virtualización, Internet de las cosas (IoT).

Unidad 3: Sistemas de Información en las Organizaciones. Concepto de sistemas de información. Tipos de sistemas de información: transaccionales, de apoyo a la decisión, de gestión empresarial. Ciclo de vida de los sistemas de información: análisis, diseño, desarrollo, implementación, mantenimiento. Aplicaciones de sistemas de información en diferentes áreas funcionales de la organización: finanzas, recursos humanos, marketing, producción.

Unidad 4: Gestión de datos en las Organizaciones. Concepto de gestión de datos. Bases de datos: estructura, diseño, modelos de datos. Sistemas de gestión de bases de datos (SGBD). Big Data y análisis de datos: conceptos básicos, herramientas y aplicaciones en las organizaciones.

Unidad 5: Seguridad de la Información. Concepto de seguridad de la información. Amenazas y vulnerabilidades en las organizaciones. Medidas de seguridad informática: autenticación, cifrado, control de accesos. Gestión de riesgos y continuidad del negocio.

Unidad 6: Tendencias y Futuro de las TI en las Organizaciones. Tendencias emergentes en Tecnologías de la Información: inteligencia artificial, blockchain, computación cuántica. Impacto de las tecnologías emergentes en las organizaciones. Desafíos y oportunidades futuras en el ámbito de las TI en las organizaciones.

6. 106. Fundamentos de Programación II

Unidad 1: Estructuras de datos avanzadas. Listas de comprensión. Diccionarios y conjuntos avanzados (comprensión de diccionarios, conjuntos y métodos avanzados). Tuplas nombradas (namedtuples) y otras estructuras de datos avanzadas de la biblioteca collections.

Unidad 2: Programación funcional en Python. Funciones lambda. Funciones de orden superior (map, filter, reduce). Uso de las funciones itertools para la programación funcional.

Unidad 3: Programación orientada a objetos (POO) avanzada. Herencia múltiple y el método de resolución de atributos (MRO). Propiedades y decoradores. Clases abstractas y métodos abstractos.

Unidad 4: Manejo avanzado de excepciones. Gestión de excepciones en contextos más complejos. Uso de los bloques try-except-finally con manejo de archivos y recursos.

Unidad 5: Programación concurrente y paralela. Introducción a la concurrencia y paralelismo. Hilos (threads) y procesos en Python. Uso del módulo threading y multiprocessing.

Manejo de archivos y serialización avanzada. Lectura y escritura de archivos binarios. Serialización y deserialización de datos en Python.

Unidad 6: Conceptos básicos de Tkinter: Fundamentos, como la creación de ventanas, widgets (como botones, etiquetas, entradas, etc.), y cómo organizar los widgets utilizando diferentes tipos de contenedores como frames y grid. Eventos y manejo de eventos. Diseño de interfaces de usuario. Estilos y temas. Gráficos y animaciones.

Unidad 7: Testing y debugging. Introducción a las pruebas unitarias con unittest. Debugging con PDB (Python Debugger). Uso de herramientas de profiling para identificar cuellos de botella en el código.

Unidad 8: Introducción a la programación web con Python. Conceptos básicos de desarrollo web (HTTP, HTML, CSS). Uso de frameworks web en Python (por ejemplo, Flask o Django). Creación de una aplicación web básica.

Introducción a la ciencia de datos con Python. Uso de bibliotecas como NumPy, Pandas y Matplotlib. Manipulación y análisis de datos. Visualización de datos.

7. 107. Análisis y Diseño de sistemas I

Unidad 1: Introducción al Análisis y Diseño de Sistemas. Conceptos fundamentales. Importancia en el desarrollo de aplicaciones. Ciclo de vida del desarrollo de sistemas: fases y metodologías. Herramientas y técnicas utilizadas en el análisis y diseño de sistemas.

Unidad 2: Requisitos del Sistema. Definición y clasificación. Técnicas de recolección de requisitos: entrevistas, encuestas, talleres. Análisis de casos de uso y escenarios de usuario. Documentación de requisitos.

Unidad 3: Modelado de Datos. Fundamentos, conceptos de entidad, atributo, relación. Modelado de datos con diagramas entidad-relación

(ER). Normalización de bases de datos. Herramientas de modelado de datos.

Unidad 4: Diseño Arquitectónico del Sistema. Arquitecturas de software: monolíticas, cliente-servidor, en capas, orientadas a servicios (SOA). Principios de diseño arquitectónico: cohesión, acoplamiento, reutilización. Patrones de diseño comunes en el desarrollo de aplicaciones. Documentación de la arquitectura del sistema: diagramas de arquitectura, descripciones técnicas.

Unidad 5: Diseño de Interfaz de Usuario (UI). Principios de diseño de interfaz de usuario: usabilidad, accesibilidad, estética. Herramientas y técnicas de diseño de UI: wireframes, prototipos, mockups. Diseño responsivo y adaptativo para diferentes dispositivos. Documentación de diseño de UI: guías de estilo, especificaciones de diseño.

Unidad 6: Diseño de la Lógica de Negocio. Definición de reglas de negocio y lógica de aplicación. Diseño de diagramas de clases y diagramas de secuencia. Implementación de patrones de diseño. Documentación de diseño de la lógica de negocio: diagramas UML, descripciones de clases y métodos.

Unidad 7: Pruebas y Validación del Diseño. Conceptos básicos, tipos de pruebas, estrategias de pruebas. Planificación y diseño de pruebas: casos de prueba, matrices de trazabilidad. Ejecución y seguimiento de pruebas: registro de errores, informes de pruebas. Validación del diseño: aseguramiento de la calidad del diseño del sistema.

Unidad 8: Herramientas y Tecnologías para Análisis y Diseño. Introducción a herramientas CASE (Computer-Aided Software Engineering). Herramientas de diseño de UI y prototipado.

8. 108. Base de Datos I

Fundamentos y Diseño de Bases de Datos relacionales

Unidad 1: Introducción a las bases de datos. Conceptos básicos de bases

de datos y sistemas de gestión de bases de datos (SGBD). Importancia de las bases de datos en el desarrollo de aplicaciones. Breve historia y evolución de las bases de datos.

Unidad 2: Modelado de datos. Conceptos de entidad, atributo, relación y clave en el modelado de datos. Diagramas de entidad-relación (DER) y su uso en el diseño de bases de datos. Normalización de bases de datos: formas normales y eliminación de anomalías.

Unidad 3: Lenguaje de manipulación de datos (DML). Consultas básicas utilizando SQL (Structured Query Language). Operaciones CRUD (Crear, Leer, Actualizar, Eliminar) en bases de datos relacionales. Filtrado de datos, ordenamiento, agrupamiento y funciones de agregación.

Unidad 4: Lenguaje de definición de datos (DDL). Creación, modificación y eliminación de tablas en una base de datos. Definición de restricciones de integridad: clave primaria, clave foránea, restricciones de unicidad, etc. Creación de índices para mejorar el rendimiento de las consultas.



Fundamentos y Tendencias

Unidad 1: Introducción a la tecnología y su impacto en la sociedad. Evolución de la tecnología y su influencia en diferentes aspectos de la vida moderna. Importancia de la innovación tecnológica en el desarrollo de aplicaciones.

Unidad 2: Tendencias tecnológicas emergentes. Inteligencia Artificial y Machine Learning: aplicaciones prácticas y tendencias futuras. Internet de las Cosas (IoT) y sus aplicaciones en diferentes industrias. Blockchain y su potencial en la transformación de la economía digital. Realidad Virtual (VR) y Realidad Aumentada (AR) en el desarrollo de aplicaciones.

Unidad 3: Metodologías ágiles de desarrollo de software. Principios de Agile y Scrum. Herramientas de gestión de proyectos ágiles como Jira, Trello, etc. Casos de estudio de empresas que aplican metodologías ágiles en su desarrollo de software.

Unidad 4: Desarrollo de habilidades blandas. Comunicación efectiva y trabajo en equipo. Resolución de problemas y toma de decisiones. Gestión del tiempo y la productividad.

Unidad 5: Seguridad informática y protección de datos. Principios básicos de seguridad informática. Amenazas comunes en el desarrollo de aplicaciones y cómo mitigarlas. Cumplimiento de regulaciones de privacidad de datos como el GDPR y la Ley de Protección de Datos Personales en Argentina.

10.110. Práctica Profesional I

Introducción a la Experiencia laboral

Unidad 1: Introducción al mundo laboral.

Expectativas y responsabilidades en un entorno profesional.

Unidad 2: Desarrollo de habilidades técnicas. Participación en un proyecto real bajo la supervisión de un mentor o tutor.

Análisis de requisitos, diseño, implementación y pruebas de una aplicación o módulo específico. Documentación del proceso.

Unidad 3: Proyecto de integración. Aplicación de conocimientos adquiridos a un proyecto real. Desarrollo de software utilizando tecnologías y herramientas relevantes para la industria.

Participación en reuniones de equipo y comunicación efectiva con colegas y superiores. Colaboración en un entorno multidisciplinario.

Retroalimentación y aprendizaje continuo.

11.201. Programación Móvil I

Unidad 1: Introducción al desarrollo móvil. Fundamentos de desarrollo de aplicaciones móviles. Introducción a las plataformas móviles: iOS, Android y multiplataforma. Herramientas y entornos de desarrollo: Xcode, Android Studio, Visual Studio, etc.

Unidad 2: Desarrollo de interfaces de usuario móvil. Diseño de interfaces de usuario adaptables a diferentes tamaños de pantalla. Uso de layouts, vistas y controles básicos. Principios de diseño de experiencia de usuario (UX) y diseño centrado en el usuario.

Unidad 3: Programación en lenguajes nativos. Desarrollo de aplicaciones nativas para iOS utilizando Swift. Desarrollo de aplicaciones nativas para Android utilizando Java o Kotlin. Implementación de funcionalidades básicas: navegación, entrada de datos, etc.

Unidad 4: Principios de persistencia de datos. Uso de bases de datos locales para almacenamiento de datos en dispositivos móviles. Introducción a SQLite y otras soluciones de almacenamiento local. Implementación de operaciones CRUD (Crear, Leer, Actualizar, Eliminar).

Unidad 5: Acceso a datos remotos. Consumo de servicios web RESTful desde aplicaciones móviles. Integración de API utilizando JSON y XML para intercambio de datos. Implementación de autenticación y autorización en aplicaciones móviles.

Unidad 6: Multimedia y sensores. Uso de cámara, micrófono y otras características multimedia. Acceso a sensores del dispositivo como GPS, acelerómetro, giroscopio, etc. Desarrollo de aplicaciones que utilicen estas funcionalidades.

Unidad 7: Notificaciones y servicios en segundo plano. Implementación de notificaciones push en aplicaciones móviles. Desarrollo de servicios en

segundo plano para tareas de larga duración. Gestión de la vida útil de la aplicación y optimización del consumo de recursos.

Unidad 8: Optimización de rendimiento. Estrategias para mejorar el rendimiento y la eficiencia de las aplicaciones móviles. Herramientas de análisis de rendimiento y detección de cuellos de botella. Optimización de la experiencia del usuario y tiempos de carga.

12.202. Inglés técnico II

Unidad 1: Gramática y estructuras de oraciones. Estructuras de preguntas y respuestas más complejas. Uso de modales: can, could, may, might, should, must.

Expansión del vocabulario técnico: términos avanzados de programación, desarrollo web y bases de datos. Terminología específica de frameworks y tecnologías populares. Vocabulario relacionado con metodologías de desarrollo ágil y herramientas de gestión de proyectos.

Unidad 2: Comprensión y producción de textos técnicos. Lectura y comprensión de textos técnicos especializados: documentación de APIs, especificaciones técnicas. Redacción de informes técnicos y documentación avanzada de proyectos. Uso de lenguaje formal y técnico en la comunicación escrita.

Unidad 3: Desarrollo de habilidades de escucha avanzadas. Escucha de conferencias y charlas técnicas sobre temas especializados en desarrollo de aplicaciones. Comprensión de discusiones y debates sobre tendencias tecnológicas actuales. Transcripción y análisis de discursos técnicos para mejorar la comprensión auditiva.

Unidad 4: Presentaciones y comunicación oral avanzada. Preparación y entrega de presentaciones técnicas sobre temas especializados en desarrollo de aplicaciones. Prácticas de expresión oral en situaciones profesionales: entrevistas de trabajo, presentaciones de proyectos. Desarrollo de habilidades de argumentación y persuasión en inglés.

Preparación para entrevistas laborales. Simulación de entrevistas

laborales en inglés para puestos relacionados con el desarrollo de aplicaciones. Desarrollo de habilidades de presentación personal y profesional en inglés.

13.203. Programación Web I

Unidad 1: Introducción al Desarrollo Web. Conceptos básicos de desarrollo web: cliente, servidor, protocolo HTTP. Estructura básica de una página web: HTML, CSS, JavaScript. Herramientas y entornos de desarrollo web: editores de código, navegadores, herramientas de depuración.

Unidad 2: HTML y CSS. Fundamentos de HTML: estructura de un documento HTML, etiquetas semánticas, formularios. Estilizado con CSS: selectores, propiedades de estilo, diseño responsive. Prácticas de diseño de interfaces web: maquetación de páginas, uso de frameworks CSS (Bootstrap).

Unidad 3: JavaScript Básico. Introducción a JavaScript: variables, tipos de datos, operadores. Estructuras de control de flujo: condicionales, bucles. Manipulación del DOM: selección de elementos, manipulación de contenido y estilo.

Unidad 4: Introducción a la Programación Orientada a Objetos en JavaScript. Conceptos: clases, objetos, herencia, polimorfismo. Uso de clases en JavaScript: definición de clases, métodos y propiedades. Aplicaciones prácticas de la programación orientada a objetos en el desarrollo web.

Unidad 5: Introducción a Frameworks de Desarrollo web. Conceptos de framework: ventajas, arquitectura MVC. Uso de frameworks backend populares: Express.js, Django, Flask. Generación de aplicaciones web rápidas y eficientes con frameworks MVC.

Unidad 6: Desarrollo de Aplicaciones de una página (SPA). Introducción a las SPAs: ventajas, desventajas. Uso de frameworks frontend populares: React.js, Angular, Vue.js. Creación de aplicaciones web dinámicas y responsivas con SPAs.

Unidad 7: Gestión de estado en Aplicaciones web. Concepto de estado en aplicaciones web: estado local, estado global. Gestión de estado con Redux (para React.js) o Vuex (para Vue.js). Implementación de flujos de datos reactivos en aplicaciones web.

Unidad 8: Despliegue de Aplicaciones web. Estrategias de despliegue: servidores web, servicios de alojamiento en la nube. Configuración de servidores web para aplicaciones web. Automatización de procesos de despliegue con herramientas como Docker y CI/CD.

14.204. Base de Datos II

Bases de datos avanzadas y Bases de datos No relacionales

Unidad 1: Optimización de consultas. Estrategias para mejorar el rendimiento de consultas SQL. Uso de índices, claves foráneas y optimizadores de consultas. Análisis de planes de ejecución y optimización de consultas.

Unidad 2: Transacciones y concurrencia. Concepto de transacción y sus propiedades ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad). Control de concurrencia: bloqueos, aislamiento de transacciones, nivel de aislamiento.

Unidad 3: Bases de datos no relacionales (NoSQL). Introducción a las bases de datos NoSQL y sus tipos (documentales, clave-valor,

columnares, grafos). Comparación entre bases de datos relacionales y NoSQL. Ejemplos de bases de datos NoSQL populares: MongoDB, Cassandra, Redis, Neo4j, etc.

Unidad 4: Uso de bases de datos NoSQL en el desarrollo de aplicaciones. Modelado de datos en bases de datos NoSQL. Consultas y operaciones básicas en bases de datos NoSQL. Casos de uso y ejemplos de aplicaciones que se benefician de bases de datos NoSQL.

Unidad 5: Integración con lenguajes de programación. Conexión y manipulación de bases de datos desde aplicaciones desarrolladas en diferentes lenguajes (Java, Python, JavaScript, etc.).

Uso de ORM (Object-Relational Mapping) para simplificar la interacción con bases de datos relacionales. Herramientas y bibliotecas populares para trabajar con bases de datos en entornos de desarrollo.

Unidad 6: Seguridad y gestión de bases de datos. Consideraciones de seguridad en el acceso a bases de datos: autenticación, autorización, roles, etc. Estrategias de copia de seguridad y recuperación de datos. Monitoreo y optimización del rendimiento de la base de datos.

15.205. Marketing de Aplicaciones I

Fundamentos y Estrategias Básicas

Unidad 1: Introducción al marketing de aplicaciones. Definición y alcance del marketing de aplicaciones. Importancia del marketing en el éxito de una aplicación.

Unidad 2: Investigación de mercado y análisis de la competencia. Métodos

para identificar el público objetivo de la aplicación. Análisis de la competencia en el mercado de aplicaciones. Herramientas y técnicas para realizar investigaciones de mercado.

Unidad 3: Desarrollo de una propuesta de valor. Identificación de los beneficios claves de la aplicación para los usuarios. Definición de la propuesta única de venta (USP). Creación de mensajes de marketing persuasivos.

Unidad 4: Estrategias de posicionamiento y branding. Elección del nombre de la aplicación y diseño del logo. Desarrollo de la identidad visual de la marca. Posicionamiento de la aplicación en el mercado y diferenciación de la competencia.

Unidad 5: Estrategias de promoción y difusión. Desarrollo de un plan de lanzamiento de la aplicación. Uso de medios de comunicación social (Redes Sociales, Influencers, etc.). Estrategias de relaciones públicas y publicidad.

Unidad 6: Optimización para tiendas de aplicaciones (ASO). Optimización de la página de la aplicación en las tiendas de aplicaciones (App Store Optimization). Uso de palabras clave relevantes y descripciones atractivas. Estrategias para aumentar las descargas orgánicas.

16.206. Programación Móvil II

Unidad 1: Desarrollo multiplataforma con frameworks. Introducción a frameworks multiplataforma como React Native, Flutter y Xamarin.

Ventajas y desventajas del desarrollo multiplataforma frente al desarrollo nativo.

Unidad 2: Arquitectura de aplicaciones móviles. Patrones de diseño para el desarrollo de aplicaciones móviles (MVC, MVVM, MVP, etc.). Separación de responsabilidades y principios SOLID en el diseño de aplicaciones móviles. Implementación de arquitecturas escalables y mantenibles.

Unidad 3: Principios de seguridad en el desarrollo de aplicaciones móviles. Mejores prácticas para proteger datos sensibles y prevenir vulnerabilidades. Criptografía, autenticación, autorización y gestión de sesiones.

Unidad 4: Desarrollo de una aplicación móvil completa desde cero hasta la implementación. Implementación de funcionalidades avanzadas y características específicas del proyecto. Integración de todas las tecnologías y habilidades aprendidas.

Unidad 5: Gestión de proyectos móviles. Planificación, seguimiento y control de proyectos móviles. Metodologías ágiles de gestión de proyectos como Scrum o Kanban. Colaboración en equipos de desarrollo y comunicación con stakeholders.

Unidad 6: Pruebas y depuración. Estrategias de pruebas de software para aplicaciones móviles. Automatización de pruebas de interfaz de usuario y pruebas de regresión. Depuración de errores y optimización de código.

Unidad 7: Despliegue y distribución de aplicaciones. Preparación de aplicaciones para su distribución en tiendas de aplicaciones (App Store, Google Play, etc.). Configuración de versiones beta y distribución interna. Monitoreo y actualización de aplicaciones en producción.

Unidad 8: Seguridad en Aplicaciones web. Aspectos avanzados de seguridad en aplicaciones web: autenticación multifactor, gestión de sesiones seguras y protección contra ataques. Implementación de políticas de seguridad, como CORS y Content Security Policy. Pruebas de

seguridad y auditorías de código para identificar y mitigar vulnerabilidades.

17. Análisis y Diseño de sistemas II

Unidad 1: Introducción a la Ingeniería de Software avanzada. Modelos de ciclo de vida de desarrollo de software. Principios de diseño SOLID y principios de diseño orientado a objetos avanzados.

Unidad 2: Desarrollo ágil de software. Metodologías ágiles: Scrum, Kanban, Extreme Programming (XP). Roles y responsabilidades en equipos ágiles: scrum master, product owner, equipo de desarrollo. Prácticas ágiles: daily stand-ups, sprint planning, revisión de sprint, retrospectivas.

Unidad 3: Modelado avanzado de bases de datos relacionales: diseño físico, optimización de consultas. Modelado de datos multidimensionales: cubos OLAP, esquemas de estrella y copo de nieve. Modelado de datos NoSQL: bases de datos documentales, de grafos, de columnas anchas. Herramientas avanzadas de modelado de datos: ERwin, Oracle SQL Developer Data Modeler.

Unidad 4: Diseño Avanzado de Interfaz de Usuario (UI). Principios de diseño de experiencia de usuario (UX) avanzados. Diseño de interfaces de usuario basadas en componentes y patrones de diseño. Consideraciones de accesibilidad y usabilidad en el diseño de interfaces de usuario. Herramientas avanzadas de diseño de UI/UX: Sketch, Figma, Adobe XD. Prototipado interactivo y pruebas de usabilidad avanzadas.

Unidad 5: Modelado y arquitectura de sistemas. Modelado de sistemas utilizando diagramas UML (Unified Modeling Language). Principios de diseño arquitectónico para aplicaciones de gran escala. Patrones arquitectónicos comunes: MVC, MVVM, MVA. Evaluación y selección de arquitecturas adecuadas para diferentes tipos de aplicaciones.

Unidad 6: Pruebas y Mantenimiento de Sistemas. Tipos de pruebas: unitarias, integración, funcionales, de aceptación, etc. Estrategias de

automatización de pruebas y herramientas populares. Pruebas de rendimiento y seguridad.

Mantenimiento y evolución de sistemas. Estrategias para la gestión de cambios y actualizaciones en sistemas en producción. Mantenimiento correctivo, adaptativo, perfectivo y preventivo. Herramientas y técnicas para la refactorización de código y la mejora continua.

18.208. Programación Web II

Unidad 1: Desarrollo Full-Stack. Introducción al desarrollo full-stack: comprensión del stack de tecnologías. Integración de front-end y back-end en aplicaciones web. Uso de frameworks y herramientas para el desarrollo full-stack, como MEAN (MongoDB, Express.js, Angular, Node.js) o MERN (MongoDB, Express.js, React.js, Node.js).

Unidad 2: Arquitectura de Microservicios. Conceptos básicos de microservicios: arquitectura, beneficios, desafíos. Diseño de sistemas basados en microservicios. Implementación de microservicios utilizando tecnologías como Docker y Kubernetes.

Unidad 3: Desarrollo de Aplicaciones Progresivas (PWA). Introducción a las Progressive Web Apps (PWA): características, ventajas. Implementación de funcionalidades de PWA, como trabajadores de servicio, instalación en el hogar y notificaciones push. Optimización de PWA para rendimiento y experiencia del usuario.

Unidad 4: Desarrollo de Aplicaciones web con tecnologías emergentes. Exploración de tecnologías y tendencias emergentes en el desarrollo web, como WebAssembly, GraphQL y WebRTC. Casos de uso y aplicaciones prácticas de estas tecnologías en el desarrollo de aplicaciones web modernas. Evaluación de las ventajas y desventajas de adoptar tecnologías emergentes en proyectos reales.

Unidad 5: Análisis de Rendimiento y Optimización. Herramientas y técnicas para analizar el rendimiento de aplicaciones web: Chrome DevTools, Lighthouse, WebPageTest. Identificación de cuellos de botella de

rendimiento y áreas de mejora. Estrategias de optimización de rendimiento, incluida la optimización de imágenes, la compresión de recursos y el almacenamiento en caché.

Unidad 6: Pruebas Automatizadas y Control de Calidad. Introducción a las pruebas automatizadas en el desarrollo web: pruebas unitarias, pruebas de integración, pruebas de extremo a extremo. Herramientas populares de pruebas automatizadas para aplicaciones web, como Jest, Selenium y Cypress. Implementación de pipelines de integración continua (CI) y entrega continua (CD) para automatizar pruebas y despliegue de aplicaciones web.

19.209. Seminario Tecnológico II

Herramientas y Frameworks

Unidad 1: Desarrollo web avanzado. Frameworks de desarrollo web como React.js, Angular y Vue.js. Herramientas de construcción y empaquetado como Webpack y Parcel. Prácticas recomendadas para el desarrollo front-end moderno.

Unidad 2: Desarrollo de aplicaciones móviles. Plataformas de desarrollo móvil: iOS, Android y multiplataforma. Frameworks y herramientas como React Native, Flutter y Xamarin. Diseño de interfaces de usuario móvil (UI) y experiencia de usuario (UX).

Unidad 3: Desarrollo de API y servicios web. Principios RESTful y arquitectura de servicios web. Desarrollo de API con Node.js, Express.js y otros frameworks. Autenticación y autorización en servicios web.

Unidad 4: Bases de datos y almacenamiento de datos. Bases de datos relacionales y no relacionales. Modelado de datos y diseño de esquemas. Implementación de acceso a datos utilizando ORM y ODM.

The logo for UPATECO (Universidad Provincial de Tecnología) is displayed in the background. It features a stylized shield with horizontal wavy lines in red, blue, and green, surrounded by a circular border of stars. To the right of the shield, the text 'UPATECO' is written in large, blue, block letters. Below it, 'UNIVERSIDAD PROVINCIAL' is written in smaller, red, block letters, and 'ADMINISTRACIÓN • TECNOLOGÍA • CALIDAD' is written in even smaller, blue, block letters.

20.210. Práctica Profesional II

Profundización y Especialización

Unidad 1: Desarrollo de proyectos avanzados. Participación en proyectos más complejos y de mayor escala. Implementación de funcionalidades avanzadas y soluciones innovadoras.

Gestión de proyectos: planificación, seguimiento y cumplimiento de plazos.

Unidad 2: Experiencia en diferentes áreas de desarrollo. Rotación por diferentes equipos o departamentos para adquirir experiencia en diferentes aspectos del desarrollo de aplicaciones (front-end, back-end, QA).

Exploración de tecnologías y herramientas especializadas.

Unidad 3: Desarrollo de habilidades de liderazgo. Liderazgo de equipos pequeños o liderazgo técnico en proyectos específicos. Toma de

decisiones, resolución de conflictos y delegación de tareas. Desarrollo de habilidades de mentoring y coaching.

Unidad 4: Colaboración con la comunidad. Participación en eventos de la comunidad tecnológica (hackathons, meetups, conferencias, etc.).

Contribución a proyectos de código abierto. Compartir conocimientos y experiencias con otros estudiantes y profesionales.



21.301. Marketing de Aplicaciones II

Estrategias Avanzadas y Analítica

Unidad 1: Retención de usuarios y fidelización. Estrategias para mejorar la retención de usuarios. Implementación de programas de fidelización y recompensas. Análisis del ciclo de vida del usuario.

Unidad 2: Monetización de aplicaciones. Modelos de negocio para aplicaciones móviles: freemium, suscripciones, publicidad, compras en la aplicación, etc. Estrategias para maximizar los ingresos sin comprometer la experiencia del usuario. Herramientas de análisis de monetización y

métricas clave.

Unidad 3: Análisis de datos y toma de decisiones basada en datos. Herramientas de análisis de aplicaciones como Google Analytics, Firebase Analytics, etc. Interpretación de métricas clave como descargas, usuarios activos, tiempo de retención, etc. Uso de datos para optimizar las estrategias de marketing y la experiencia del usuario.

Unidad 4: Marketing de aplicaciones en diferentes plataformas. Estrategias específicas para la promoción de aplicaciones en iOS y Android. Consideraciones especiales para aplicaciones en diferentes sectores (juegos, educación, productividad, etc.). Estrategias de marketing para aplicaciones web progresivas (PWA).

Unidad 5: Gestión de la reputación y el servicio al cliente. Monitoreo y respuesta a las reseñas de los usuarios. Gestión de la reputación online de la aplicación. Estrategias para mejorar la satisfacción del cliente y resolver problemas de manera efectiva.

Unidad 6: Estrategias de marketing internacional. Adaptación de estrategias de marketing para diferentes mercados y culturas. Localización de la aplicación: traducción de contenido, ajustes culturales, etc. Consideraciones legales y regulatorias en la promoción de aplicaciones en diferentes países.

22.302. Inglés técnico III

Unidad 1: Comunicación Avanzada en el Ámbito Profesional. Habilidades avanzadas de expresión oral en situaciones profesionales: presentaciones técnicas, negociaciones, reuniones. Desarrollo de estrategias de comunicación efectiva en entornos laborales internacionales. Prácticas de escucha activa y respuesta en situaciones de trabajo en equipo.

Unidad 2: Redacción técnica y documentación avanzada. Elaboración de documentos técnicos complejos: manuales de usuario, especificaciones técnicas, informes de proyectos. Uso de lenguaje técnico y formal en la redacción de correos electrónicos, memorandos y otros documentos de comunicación empresarial. Revisión y edición de textos para mejorar la claridad, coherencia y cohesión.

Unidad 3: Comprensión de lectura avanzada. Lectura y análisis de textos especializados en desarrollo de aplicaciones: artículos científicos, blogs técnicos, casos de estudio. Comprensión de instrucciones y tutoriales avanzados para el desarrollo de software. Evaluación crítica de información técnica y fuentes de referencia en inglés.

Unidad 4: Presentaciones y demostraciones Técnicas. Preparación y entrega de presentaciones técnicas avanzadas sobre proyectos de desarrollo de aplicaciones. Prácticas de demostraciones de software y presentación de prototipos ante audiencias técnicas y no técnicas. Desarrollo de habilidades de persuasión y argumentación en presentaciones profesionales.

Unidad 5: Prácticas profesionales y preparación para el mundo laboral. Simulación de situaciones laborales reales: reuniones de equipo, presentaciones de proyectos, interacciones con clientes. Desarrollo de habilidades de liderazgo y trabajo en equipo en entornos de desarrollo de aplicaciones. Preparación para entrevistas de trabajo en el campo de la tecnología y desarrollo de aplicaciones.

Unidad 1: Introducción a la ética y la legislación laboral. Definición de ética profesional y su importancia en el desarrollo de aplicaciones. Concepto de legislación laboral y su relación con el entorno laboral de los programadores.

Unidad 2: Derechos laborales en Argentina y a nivel internacional. Derechos laborales fundamentales reconocidos por la Organización Internacional del Trabajo (OIT). Marco legal argentino: Ley de Contrato de Trabajo y sus modificaciones, Ley de Riesgos del Trabajo, etc.

Unidad 3: Ética en el desarrollo de aplicaciones. Responsabilidad ética de los desarrolladores en la creación de software. Principios éticos en el diseño y desarrollo de aplicaciones: privacidad, accesibilidad, seguridad y transparencia.

Unidad 4: Diversidad e inclusión en el entorno laboral. Importancia de la diversidad de género, cultural y de pensamiento en equipos de desarrollo. Legislación antidiscriminatoria en Argentina y otros países. Estrategias para fomentar la inclusión y la equidad en el lugar de trabajo.

Unidad 5: Salud y seguridad laboral en el desarrollo de aplicaciones. Riesgos específicos en el trabajo de desarrollo de software: ergonomía, fatiga visual, estrés, entre otros. Legislación sobre condiciones de trabajo seguras y saludables.

Unidad 6: Propiedad intelectual y derechos de autor. Conceptos básicos sobre propiedad intelectual y derechos de autor. Legislación argentina y tratados internacionales sobre derechos de autor. Uso ético y legal de software de terceros y código abierto.

Unidad 7: Responsabilidad social empresarial (RSE) en la industria tecnológica. Compromiso de las empresas de tecnología con la sociedad y el medio ambiente. Impacto social de las tecnologías de la información y la comunicación (TIC).

Unidad 8: Aspectos legales y éticos en el trabajo remoto y la contratación freelance. Normativas y regulaciones para el trabajo remoto en Argentina y otros países. Contratos laborales y acuerdos de prestación de servicios

como freelancer. Ética y responsabilidad en la gestión del tiempo y la comunicación en entornos virtuales.

24.304. Seminario Tecnológico III

Proyectos prácticos y Aplicaciones avanzadas

Unidad 1: Desarrollo de proyectos prácticos

Aplicación de los conocimientos adquiridos en proyectos reales. Trabajo en equipos multidisciplinarios para el desarrollo completo de aplicaciones.

Unidad 2: Optimización y rendimiento de aplicaciones:

Estrategias para mejorar la velocidad y eficiencia de las aplicaciones.
Herramientas de análisis de rendimiento y detección de cuellos de botella.
Optimización de recursos y gestión de memoria.

Unidad 3: Despliegue y mantenimiento de aplicaciones:

Estrategias de despliegue en entornos de desarrollo, pruebas y producción. Automatización de procesos de despliegue. Monitoreo y gestión de aplicaciones en producción.

Unidad 4: Evaluación de nuevas tecnologías y tendencias:

Análisis crítico de nuevas tecnologías y su aplicabilidad en diferentes contextos. Evaluación de riesgos y beneficios al adoptar nuevas herramientas y frameworks. Adaptación a cambios tecnológicos y actualización continua de habilidades.

25.305. Práctica Profesional III

Preparación para el mundo laboral

Unidad 1: Experiencia en entornos laborales reales:

Pasantías o prácticas profesionales en empresas del sector tecnológico. Aplicación de habilidades y conocimientos adquiridos en un entorno de trabajo real. Exposición a procesos y metodologías de trabajo utilizados en la industria.

Unidad 2: Desarrollo de un portafolio profesional

Creación de un portafolio que destaque proyectos realizados durante la carrera y en prácticas profesionales. Documentación de contribuciones a proyectos de código abierto, participación en eventos, etc. Preparación de materiales para entrevistas de trabajo: currículum, carta de presentación, perfil en LinkedIn, etc.

Unidad 3: Preparación para la transición al mercado laboral:

Asesoramiento sobre búsqueda de empleo, entrevistas y negociación salarial. Desarrollo de habilidades de networking y establecimiento de contactos en la industria. Evaluación de opciones de carrera y planificación profesional a largo plazo.

11.- METODOLOGÍA DE ENSEÑANZA

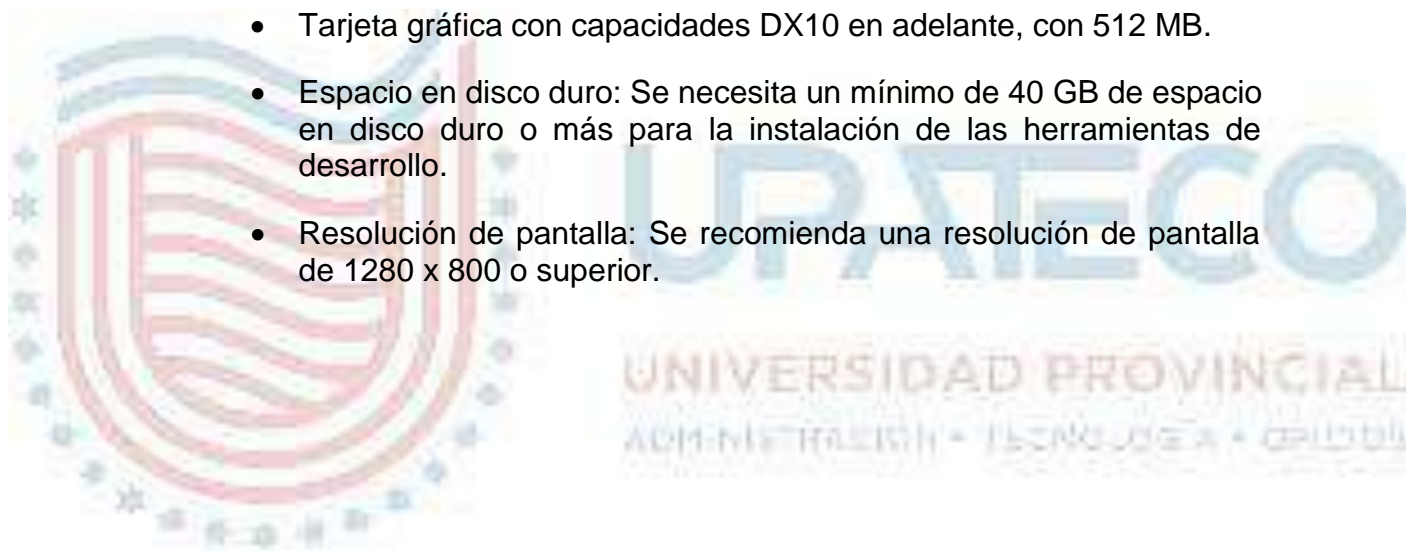
La estrategia de enseñanza se llevará a cabo de manera exclusivamente virtual con el objetivo de facilitar la participación de los estudiantes en áreas remotas y ofrecer una mayor flexibilidad de horarios para aquellos que tienen compromisos laborales u otras responsabilidades.

Se programarán sesiones de clases tanto sincrónicas, utilizando tecnologías para la interacción en tiempo real, como asincrónicas. En el caso de las sesiones sincrónicas, se grabarán y estarán disponibles en la plataforma para aquellos que no puedan asistir en el momento programado. Este enfoque garantizará horarios adaptables para atender a todas las personas interesadas en adquirir conocimientos sobre las herramientas esenciales para diseñar y desarrollar aplicaciones. Se plantea la realización de videoconferencias para las clases sincrónicas.

El acceso a los materiales de estudio y a las diversas actividades se llevará a cabo mediante una plataforma de aprendizaje virtual. Esta plataforma ofrecerá espacios de videoconferencia para las clases en tiempo real. Además, se hará uso de los recursos educativos integrados en la plataforma, permitiendo a los estudiantes acceder de manera continua y asincrónica al material pedagógico. Estos recursos incluirán foros de discusión, áreas para la realización de tareas, wikis y otras actividades que facilitarán la interacción entre estudiantes y docentes. También se implementarán cuestionarios automatizados en línea, entre otros recursos, para brindar oportunidades adicionales de aprendizaje y evaluación.

Requisitos para cursar esta carrera: el o la estudiante que quiera cursar esta carrera debe disponer de un equipo informático con las siguientes características mínimas para poder seguir adecuadamente el desarrollo de los contenidos propuestos.

- PC o notebook con conexión a internet estable de por lo menos 10 Mbps.
- Sistema operativo: Windows (7/8/10), macOS o Linux.
- Procesador: Un procesador compatible con la virtualización de hardware, como Intel® Core™ i5 o superior.
- Memoria RAM: Se recomienda un mínimo de 8 GB de RAM, aunque se prefiere 16 GB o más para un rendimiento óptimo.
- Tarjeta gráfica con capacidades DX10 en adelante, con 512 MB.
- Espacio en disco duro: Se necesita un mínimo de 40 GB de espacio en disco duro o más para la instalación de las herramientas de desarrollo.
- Resolución de pantalla: Se recomienda una resolución de pantalla de 1280 x 800 o superior.



12.- FORMA DE EVALUACIÓN

Los Laboratorios de Certificación se realizarán a través de videoconferencia para los estudiantes del interior de la Provincia y en forma presencial para los estudiantes de Salta Capital. Cabe destacar que es requisito indispensable para poder cursar esta tecnicatura que el/la estudiante disponga de una PC con una conexión a internet estable, tanto para el acceso a los recursos, como a las clases sincrónicas, la realización de los trabajos prácticos y el Laboratorio final.

