

Descripción de la Certificación para Analista de Conocimiento-Dimensión Programador

I. Identificación de la certificación

- I.1. *Sector/es de actividad socio productiva:* **Informática - Software y Servicios Informáticos**
- I.2. *Denominación del perfil profesional:* **PROGRAMADOR**
- I.3. *Familia profesional:* **INFORMATICA.**
- I.4. *Denominación del certificado de referencia:* **PROGRAMADOR**
- I.5. *Ámbito de la trayectoria formativa:* **FORMACIÓN PROFESIONAL.**
- I.6. *Tipo de certificación:* **CERTIFICADO DE FORMACIÓN PROFESIONAL INICIAL**
- I.7. *Nivel de la Certificación* **III**

II. Referencial al Perfil Profesional

El perfil profesional del Programador alude al conjunto de actividades en las que el profesional puede demostrar, una vez que ha logrado su certificación, los conocimientos adquiridos sobre su profesión.

Este perfil involucra un conjunto de competencias específicas de desarrollo de software dentro del sector profesional de Informática. Se articula en torno a núcleos curriculares a partir de las demandas socio-productivas del sector y la realidad del medio industrial y de servicios.

II.1.-*Alcance del perfil profesional*

El *Programador* está capacitado, de acuerdo a las actividades que se detallan en el Perfil Profesional, para escribir, testear, depurar y mantener el código de los programas en el marco de un equipo de trabajo o en forma individual.

Los programadores escriben los programas de acuerdo a las especificaciones dadas por los analistas de sistemas. Ellos transforman el diseño en instrucciones concretas que pueden ejecutarse en la computadora.

También tienen como tarea verificar y validar el programa en el contexto del proceso de prueba y depuración de errores.

El software necesita ser actualizado, corregido, modificado y ampliado, es decir mantenido, siendo esta tarea realizada en parte por los programadores.

Este profesional trabaja integrado en un equipo y coordinado por un Líder de Proyecto y se remite a este para obtener información sobre lo que se espera de él.

Funciones que ejerce el profesional

A continuación, se presentan funciones y sub-funciones del perfil profesional del Programador:

1. Construir código de programación de acuerdo a especificaciones.

El Programador interpreta las especificaciones de diseño y de requisitos de las asignaciones a programar, comprendiendo en su contexto inmediato, cuál es el problema a resolver, determinando el alcance del problema y convalidando su interpretación a fin de identificar aspectos faltantes.

Desarrolla algoritmos que den soluciones a los problemas asignados o derivados de los mismos, procurando tener un código eficiente, documentado, fácil de entender y mantener. Efectúa pruebas de unidad al código construido para asegurar que cumpla con las especificaciones recibidas.

2. Interpretar especificaciones de diseño que le permitan construir el código en el contexto del desarrollo de software en el que participa.

El Programador recibe las especificaciones y analiza el problema a resolver; interpreta el material recibido y clarifica eventuales malas interpretaciones o desacuerdos convalidando su interpretación con los miembros del equipo de proyecto que correspondan.

Debe ser analítico y tener capacidad de abstracción, para poder comprender las especificaciones, observando reglas de los lenguajes de modelado en la que estas especificaciones están expresados. También deberá describir en sus propios términos el problema, identificar puntos ambiguos, aspectos faltantes o eventuales contradicciones entre distintos requisitos a cumplir o inconsistencias entre estos y otros aspectos conocidos del producto.

3. Dimensionar su trabajo en el contexto del proyecto de desarrollo de software.

El Programador como parte integrante de un equipo de proyecto debe poder estimar el esfuerzo que necesita para realizar un trabajo que le fue asignado. Para ello deberá procurarse la información que necesite para dimensionar el trabajo, considerando la utilización de recursos de los que disponga para ser productivo, por ejemplo, utilización de bibliotecas de componentes, aplicación de patrones, entre otros.

4. Verificar el código desarrollado.

El Programador determina las necesidades de cobertura de las pruebas, en función de las características y definiciones de calidad definidas para el producto, identifica las clases de equivalencia de datos utilizados internamente o intercambiados y ejecuta los casos de prueba.

Realiza las pruebas correspondientes, registrando los datos y resultados alcanzados, así como las acciones correctivas realizadas para solucionar los defectos encontrados.

5. Depurar estructuras lógicas o códigos de programas

El Programador relaciona resultados incorrectos con los datos o porciones de código que los originaron, analiza estos datos y/o partes del código que causaron el mal funcionamiento y determina el tipo de corrección o reemplazo requeridos; verifica que la corrección y/o reemplazo solucionen el mal funcionamiento o la salida de resultados erróneos.

6. Utilizar estructuras de datos vinculadas con las aplicaciones desarrolladas o a desarrollar.

El Programador conoce la estructura de los datos, su organización, la relación entre entidades y su uso en las aplicaciones desarrolladas o por desarrollar.

También consulta a pares y al líder del equipo de proyecto para reflexionar y recibir ayuda que le permita resolver problemas relacionados con el manejo de los datos. También aporta sus conocimientos a otros.

7. Elaborar documentación técnica de acuerdo con los requerimientos funcionales y técnicos recibidos.

El Programador realiza la documentación describiendo qué hace cada parte del código y por qué se incluyen, qué datos o situación lo originaron; registros y evidencias de las actividades realizadas y de los incidentes observados, debiendo identificar cada versión de acuerdo a estándares establecidos dentro del equipo de desarrollo.

8. Integrar un equipo en el contexto de un Proyecto de Desarrollo de Software.

El desarrollo de software es una actividad social, que se desarrolla principalmente en equipo, en consecuencia, el Programador debe poder integrarse en un equipo de trabajo, sea este un contexto de proyecto de gestión tradicional o de gestión ágil.

Debe poder manejar su entorno personal y el entorno laboral donde se insertará.

Área Ocupacional

Este profesional se inserta ocupacionalmente en distintos tipos de organizaciones donde se realiza desarrollo de software, ya sea por requerimiento de otras organizaciones locales o extranjeras o por solicitud de usuarios particulares.

Entre dichas organizaciones se cuentan empresas que proveen software conjuntamente con servicios de asesoramiento y consultoría, y que desarrollan sus propios productos de software para vender en el país o en el exterior. También en organizaciones dedicadas a otras actividades, pero que producen el software que necesitan para desarrollar sus propias actividades o que integran en productos que venden.

El software debe satisfacer las especificaciones de los requerimientos, ya sean formales o informales, las que pueden venir dadas por el cliente, algún consultor especializado en el tipo de problemas que aborda la aplicación o ser desarrollada por algún analista funcional integrante del equipo de trabajo del proyecto.

El equipo de desarrollo suele estar encabezado por un jefe o líder de proyecto, quien es responsable por el proyecto. Integra diversos roles ocupacionales, como el de arquitecto de software, que establece el diseño global del sistema y las especificaciones de calidad de la solución, así como una serie de programadores que son quienes realizan la construcción de acuerdo a las especificaciones recibidas.

Del equipo de trabajo pueden participar uno o más analistas técnicos que se ocupan de detalles relativos a aspectos de tecnología, seguridad, bases de datos o estándares de programación y asesoran y dan apoyo técnico a los programadores. Eventualmente pueden participar diseñadores gráficos y especialistas en otros aspectos específicos.

La posición ocupacional de la figura que es referencia del presente trayecto es la que suele denominarse Programador. Integra equipos de proyecto dedicados al desarrollo o mantenimiento de software y recibe

asignaciones específicas que debe resolver en lapsos que suelen medirse en términos de días o semanas, produciendo módulos o programas que satisfagan especificaciones y se integren al sistema objeto del proyecto.

A partir de especificaciones de requerimientos y diseño, y del conocimiento de la arquitectura del sistema, los programadores construyen la parte que les fuera asignada en base a desarrollos ya existentes y adaptando o escribiendo nuevo código. Testean su trabajo, verifican unitariamente lo producido y lo entregan para ser probado integralmente e integrado al resto.

Habitualmente pueden estar especializados en una tecnología determinada, trabajan individualmente o de a pares dentro de un grupo más numeroso, brindándose mutuamente colaboración para resolver los problemas que deben enfrentar.

También puede desempeñarse en forma autónoma, asumiendo la mayor parte de las tareas propias del proceso, sobre todo trabajando en forma independiente resolviendo problemas de pequeñas organizaciones que requieren sistemas de baja complejidad y reducida dimensión.

De lo anterior se desprende que el *Programador* desarrolla su actividad en las siguientes áreas ocupacionales:

- Servicios informáticos para pequeñas y medianas empresas en áreas de desarrollo y producción de software.
- Empresas de distintos sectores de actividad económica en áreas de informática o de procesamiento de datos.
- Por cuenta propia o en pequeños emprendimientos asociativos de desarrollo y producción de software.
- Empresas de servicios de mantenimiento de sistemas informáticos.
- Comercialización de equipos y sistemas informáticos.
- Administración pública, en las áreas de mantenimiento y gestión de la información
- ONGs, en áreas vinculadas con el procesamiento de datos para la gestión.
- Mantenimiento de sistemas informáticos en entornos personales
- Asesoramiento técnico y venta de sistemas y aplicaciones informáticas.

III. Trayectoria Formativa del Programador.

III.1. Las capacidades profesionales del Programador

El proceso de formación habrá de organizarse en torno a la adquisición y acreditación de un conjunto de capacidades profesionales que se corresponden con los desempeños descritos en el Perfil Profesional.

Capacidades profesionales para el perfil en su conjunto

- Interpretar información, escrita o verbal, relacionada con especificaciones de requerimientos y de diseño, identificando notaciones propias de la actividad, verificando su pertinencia y alcance para realizar una acción requerida.

- Transferir la información de los documentos a su tarea, relacionada con desarrollo de software, verificando su pertinencia y alcance para realizar una acción requerida a partir de la especificación de diseño recibida.
- Identificar los problemas que se presenten en el desarrollo o programación a partir del análisis, jerarquización y priorización de la información.
- Integrar las técnicas de trabajo, la información, la reutilización de códigos existentes, los criterios de calidad de software en las actividades de desarrollo propias de su actividad.
- Aplicar criterios de calidad de software (cohesión y acoplamiento) en los procesos de desarrollo de software tendiendo a generar propuestas de mejoramiento continuo en métodos de programación, técnicas de prueba unitaria y organización del trabajo en equipo.
- Establecer relaciones de cooperación, coordinación e intercambio en el propio equipo de trabajo y con otros equipos de desarrollo de software que intervengan con el equipo de trabajo.
- Transmitir información técnica de manera verbal, sobre el desarrollo de piezas de software, que le fueron encomendadas.
- Gestionar y administrar sus propios recursos (principalmente tiempos), necesarios para el avance de su propio trabajo y el trabajo en equipo, según las condiciones establecidas por los responsables de las tareas encomendadas.
- Gestionar la relación comercial que posibilite la obtención de empleo y las relaciones que devengan con los prestadores de servicios.

III.2. Estructura modular del trayecto curricular de la figura profesional del Programador

La estructura modular del trayecto curricular de esta figura profesional, se organiza en base a una serie de módulos comunes, de base y de gestión que guardan correspondencia con el campo científico – tecnológico y módulos específicos que corresponden al campo de formación técnico específico y de las prácticas profesionalizantes. A saber:

Módulos	Horas reloj
Técnicas de Programación	120 hs.
Desarrollo de Software	24 hs.
Programación Orientada a Objetos	150 hs
Base de Datos	70 hs
Total de Horas de Formación Común, Científica tecnológica, - Módulos comunes.	120
Total, de horas de formación técnica específica y prácticas profesionalizantes- Módulos específicos.	244
Total de Horas del trayecto curricular	364 hs.

III.3. Régimen pedagógico de cursado del trayecto curricular de la figura profesional Programador.

A continuación, se presenta el régimen pedagógico de cursado del trayecto curricular de la figura profesional del Programador, el mismo se organiza como una estructura de dos tramos con el objeto de clarificar el esquema posible de composición, secuencia y organización curricular de los módulos del trayecto, tanto a nivel de las correlatividades como de las opciones organizacionales posibles por parte de los CFP que ofertan este trayecto.

Descripción y síntesis del Régimen pedagógico de cursado:

- La trayectoria se inicia con el cursado obligatorio del módulo de base “Técnicas de Programación”.
- El módulo común de “Desarrollo de Software” puede dictarse en la secuencia de dos formas posibles: al inicio con el módulo de base de “Técnicas de Programación”, o bien con el módulo técnico específico o como último módulo.
- El Módulo específico “Base de Datos” puede dictarse de dos formas posible: en simultáneo con “Técnicas de Programación” o bien posterior al mismo.
- Para el cursado del módulo específico de “Programación Orientada a Objetos” es necesario haber cursado el módulo de base “Técnicas de Programación” y el módulo específico “Base de Datos”.
- Se obtiene la certificación del trayecto de Programador previa certificación de los módulos específicos “Programación Orientada a Objetos” y “Base de Datos”.

Ejemplos de cursado por tramos:

Tramo	Ejemplo 1	Ejemplo 2	Ejemplo 3	
1º	* Técnicas de Programación * Base de Datos	* Técnicas de Programación * Desarrollo de Software	* Técnicas de Programación	* Técnicas de Programación
2º	* Programación orientada a objetos * Desarrollo de Software	* Base de Datos	Ver alternativas	Ver alternativas
3º		* Programación orientada a		

		objetos		
--	--	---------	--	--

Cada uno de los módulos que integran el trayecto se acreditan mediante certificación de módulo. Con la certificación del conjunto de los módulos se accede a la certificación de Formación Profesional Inicial de nivel III de Programador.

III.4. Prácticas formativas profesionalizantes:

En relación a las **prácticas formativas profesionalizantes**, el diseño curricular del trayecto de Programador define un conjunto de prácticas formativas que se deben garantizar a partir de un espacio formativo adecuado, con todos los insumos necesarios y simulando un ambiente real de trabajo para mejorar la significatividad de los aprendizajes. La institución educativa podrá optar, mediante acuerdos con otras organizaciones socio productivas del sector profesional, el desarrollo de prácticas formativas de carácter profesionalizante en el ámbito externo a la institución educativa. En todos los casos las prácticas deben ser organizadas, implementadas, supervisadas pedagógicamente y evaluadas por el centro de formación profesional y estarán bajo el control de la propia institución educativa y del Ministerio de Educación, quien a su vez certificará la realización de las mismas.

Las prácticas pueden asumir diferentes formatos, pero sin perder nunca de vista los fines formativos que se persigue con ellas. La implementación y desarrollo del trayecto de Programador deberá garantizar la realización de las prácticas profesionalizantes definiendo en el diseño curricular los recursos necesarios para las mismas. Dichas prácticas resultan indispensables para poder evaluar las capacidades profesionales definidas en cada módulo formativo.

III.5 Trayecto Curricular: Definición de módulos

Denominación de Módulo: **Técnicas de Programación**

Tipo de Módulo: de Base

Carga Horaria: **120** hs. reloj

Carga Horaria de Práctica Profesionalizante: **80 horas Reloj**

Presentación:

El módulo Técnicas de Programación tiene, como propósito general, contribuir a que los estudiantes tengan una formación que los inicia en la construcción de las capacidades técnicas en torno a la lógica de programación.

Este módulo se constituye, así en un espacio de formación que permite a los estudiantes aproximarse a saberes que los vinculan con la formación de la figura del “Programador”.

Se parte conceptualmente de analizar problemas de base informática, con el fin que los estudiantes adquieran los conceptos y las técnicas de resolución de problemas, cuya lógica se utiliza y amplía en el módulo específico de Programación Orientada a Objetos.

En este contexto se entiende por Programación al método de resolución de problemas que utiliza algoritmos y un lenguaje de programación, respetando los principios del desarrollo de software que apuntan a la calidad, utilizados en el campo profesional de actuación de éste técnico. La calidad en el desarrollo del software implica la independencia funcional, niveles de alta cohesión y bajo acoplamiento, que posibilitan la reutilización y el manejo de la complejidad.

Para la organización de la enseñanza de esta unidad curricular se han organizado los contenidos en tres bloques:

El bloque de **elementos informáticos** se centra en la lógica de la estructura de los datos; este bloque es fundante conceptualmente, tiene por objeto el tratamiento de los contenidos básicos esenciales que se utilizan en la escritura de algoritmos eficientes.

El bloque **diseño de algoritmos** aborda las técnicas de resolución de problemas computacionales o de programación, tomando como punto de partida problemas con cierto nivel de complejidad, para en un segundo momento, descomponerlos en problemas de menor complejidad reformulándolo en términos de un problema ya conocido.

El bloque **Desarrollo de Programas** presenta como núcleo central la elaboración y construcción de programas mediante la traducción o codificación de los algoritmos en un lenguaje adecuado al paradigma.

La organización del desarrollo curricular, que se presenta en este documento, sigue una secuencia que toma como punto de partida a las estructuras de datos que se adaptan al problema a resolver, posteriormente se diseñan y representan los algoritmos que lo resuelven y por último se codifican estos algoritmos en un lenguaje de programación, dando como resultado el programa.

El propósito general de esta unidad curricular, es que los estudiantes desarrollen competencias y habilidades, y adquieran conocimientos para resolver problemas e implementar sus soluciones en un lenguaje de programación de propósito general, logrando código legible y mantenible.

Este módulo se orienta al desarrollo de las siguientes **capacidades profesionales referidas al perfil profesional en su conjunto**:

- Interpretar las especificaciones de diseño y requisitos de las asignaciones a programar
- Comprender en su contexto inmediato cuál es el problema a resolver
- Determinar el alcance del problema y convalidar su interpretación a fin de identificar aspectos faltantes.
- Desarrollar algoritmos que dan soluciones a los problemas asignados o derivados de los mismos.

En relación a las **prácticas formativas de carácter profesionalizante**, se definen como unos de los ejes estratégicos de la propuesta pedagógica para el ámbito de la FP, el situar al participante en los ámbitos

reales de trabajo con las problemáticas características que efectivamente surgen en la planificación de procedimientos o secuencias de actividades asociada al desarrollo de algoritmos y la resolución de problemas de base computacional, y que se organiza a nivel de cada módulo formativo.

Para el caso del presente modulo las **prácticas formativas profesionalizantes** y los objetivos de aprendizajes se organizan para el desarrollo de:

- Práctica de resolución de una situación problemática, real o simulada de acuerdo a especificaciones de diseño, desarrollando algoritmos que den solución a problemas específicos.

Bloques de contenidos	Prácticas Formativas Profesionalizantes
<ul style="list-style-type: none"> • Visión de Sistemas- Visión de Procesos. • Visión de Sistema de Información. Ciclo de vida del Sistema de Información. Metodologías para su desarrollo. • Concepto de Intérpretes y compiladores. Criterios de diseño y de implementación de lenguajes de programación. Nociones básicas de semántica formal. • Algoritmo: secuencia ordenada de pasos que resuelve un problema concreto. • Programa. Lenguajes de Programación. • Implementación del algoritmo en un lenguaje de programación. • Definición y análisis de problemas del campo informático. <ul style="list-style-type: none"> - Datos de entrada y salida, relación entre ellos - Variables y constantes Operadores relacionales y lógicos Operadores aritméticos Concepto de algoritmo • Desarrollo de algoritmos: Técnicas de diseño de algoritmos Herramientas para diseño de algoritmos 	<ul style="list-style-type: none"> • Abordaje del concepto de hardware y software. • Algoritmos y Programas. • Abordaje del concepto de software libre referido a las cuatro libertades de los usuarios: usar el programa con cualquier propósito; estudiar el funcionamiento del programa y adaptarlo a las necesidades; distribuir copias y mejorar el programa y hacer públicas las mejoras. El software propietario: referido a su prohibición para su uso, redistribución o modificación, o que requiere permiso expreso del titular del software. • Conceptualización de software libre en cuanto a la disponibilidad de las especificaciones técnicas, el modo en que el usuario puede crear una réplica o intervenirlo, y la libertad de la documentación y del software (los controladores) que lo acompañan y que permiten su utilización. • Abordaje del concepto de proyecto de desarrollo de software para contextualizar. • Abordaje del concepto de algoritmo como resultante de la sistematización del problema. • Selección y resolución de problemas de baja complejidad del campo informático, incrementando luego el grado de complejidad hasta localizar un dato en un arreglo enfatizando en la complejidad del problema. • Diseño del algoritmo desde lo más

<ul style="list-style-type: none"> • Resolución de Problemas y Algoritmos • Estructuras de Control. Recursividad. Eventos. Excepciones. Concurrency. • Estructuras de Datos. Tipos de Datos: Abstractos, recursivos. Representación de Datos en Memoria. Estrategias de implementación. Manejo de Memoria en Ejecución. • Algoritmos fundamentales, recorrido, búsqueda, ordenamiento y actualización. • Pilas, Colas y Listas Diagramas • Estructuras de programación <ul style="list-style-type: none"> • Estructura secuencial: <ul style="list-style-type: none"> ○ Características. ○ Definición de variables • Estructura alternativa <ul style="list-style-type: none"> ○ Características. ○ Expresiones lógicas • Estructura repetitiva o iterativa <ul style="list-style-type: none"> ○ Características. ○ Fases de un programa iterativo • Pruebas de escritorio y depuración 	<p>genérico a lo más particular por medio de sucesivos refinamientos y mejoras.</p> <ul style="list-style-type: none"> • Abordaje de la resolución de problemas de creciente complejidad mediante algoritmos eficientes. • Abordaje de las estructuras de programación iniciando por la secuencial, entendiendo a este tipo de estructura como fundamento para mejorar la comprensión de las demás. • Combinación de estructuras de programación. • Armado y comprobación de prueba de escritorio de acuerdo al grado de dificultad del algoritmo sobre el que se realiza la prueba. • Asociación del concepto de codificación con el concepto de algoritmo con el fin de resolver situaciones problemáticas por medio de la computadora. • Uso de lenguajes gráficos en la etapa de programación/codificación, para crear animaciones, juegos interactivos, etc. <p>El uso de lenguajes gráficos para este nivel se fundamenta en que por sus características (representación de los bloques de construcción de los programas por códigos de color) reducen la exigencia sobre la sintaxis de la programación, permitiendo a los estudiantes enfocarse en los problemas a resolver. Los lenguajes más estructurados requieren altos grados de abstracción (pensamiento lógico formal) y se deberán desestimar su uso para este momento de la escolarización. Así mismo, la programación por medio de interfaces gráficas permite que los estudiantes comprendan los conceptos centrales de computación, tales como: iteración, condicionales, variables, tipos de datos, eventos y procedimientos.</p>
---	---

Denominación del Módulo: **Desarrollo de Software**

Tipo de Módulo: Específico

Carga Horaria: **24 horas Reloj**

Carga Horaria de Prácticas formativas de carácter profesionalizante: **12 horas Reloj**

Presentación:

El módulo específico de **Desarrollo de Software** tiene como propósito general, contribuir a la formación de los estudiantes del ámbito de la Formación Profesional en sujetos que se integrarán a equipos de desarrollo, dado que el desarrollo de software es una actividad esencialmente de trabajo en equipo.

Este módulo selecciona un conjunto de conocimientos vinculados con la disciplina de Ingeniería de Software, particularmente en los aspectos de gestión y soporte de los proyectos de desarrollo de software.

Este módulo se orienta al desarrollo de las siguientes **capacidades profesionales, estando estas articuladas con las funciones que se describen en el alcance del perfil profesional:**

- **Integrar un equipo en el contexto de un Proyecto de Desarrollo de Software.**

El desarrollo de software es una actividad social, que se desarrolla principalmente en equipo, en consecuencia, el Programador debe poder integrarse en un equipo de trabajo, sea este un contexto de proyecto de gestión tradicional o de gestión ágil.

Debe poder manejar su entorno personal y el entorno laboral donde se insertará.

- **Dimensionar su trabajo en el contexto del proyecto de desarrollo de software.**

El Programador como parte integrante de un equipo de proyecto debe poder estimar el esfuerzo que necesita para realizar un trabajo que le fue asignado. Para ello deberá procurarse la información que necesite para dimensionar el trabajo, considerando la utilización de recursos de los que disponga para ser productivo, por ejemplo, utilización de bibliotecas de componentes, aplicación de patrones, entre otros.

Para el presente módulo, y desde el punto de vista del **contenido de la formación**, se define para el agrupamiento, la selección y el tratamiento de los contenidos, los siguientes bloques:

- **Bloque Disciplinas implicadas en el Desarrollo de Software**
- **Bloque de Gestión de Proyectos**

En el bloque **Disciplinas implicadas en el Desarrollo de Software** se focaliza en introducir los conceptos fundamentales que conforman la problemática del desarrollo de software. A partir de ellos se abordarán conceptos sobre la ingeniería de software y sus disciplinas constitutivas. Se profundizará sobre actividades vinculadas con la calidad del software tanto en etapas de construcción, como son las revisiones técnicas, como para control de calidad, como es el testing.

El bloque **Gestión de Proyectos** abordará conceptos clave para el logro exitoso de un software, vinculados a la planificación y el monitoreo de los proyectos que los desarrollan. Se presentarán los dos

enfoques principales que conviven en la industria: la gestión tradicional y la gestión ágil. Se desarrollarán los conceptos clave que sustentan cada enfoque.

En relación a las **prácticas formativas de carácter profesionalizante**, se definen como unos de los ejes estratégicos de la propuesta pedagógica para el ámbito de la FP, el situar al participante en los ámbitos reales de trabajo con las problemáticas características de desempeño ocupacional/profesional.

Las prácticas formativas que se proponen para este módulo se organizan en torno a la integración de los participantes en un proyecto de desarrollo de software y que puedan desarrollar actividades vinculadas con la construcción de un producto de software, con algún rol específico, asignado. La expectativa sobre este tipo de prácticas es lograr que la vivencia los aproxime a la realidad de la problemática de la industria de software.

Los objetivos de aprendizaje a tener en cuenta para la evaluación al finalizar el cursado del módulo de “Desarrollo de Software” serán:

- Identificar las disciplinas que conforman la Ingeniería de Software y las técnicas y herramientas relacionadas.
- Conocer los tipos de procesos y los modelos de procesos más adecuados para el desarrollo de software en cada situación particular.
- Introducir los enfoques de gestión de proyectos tradicional y ágil.
- Conocer los principales métodos de desarrollo y gestión ágil.
- Valorar la relación existente entre el Proceso, el Proyecto y el Producto de Software a construir
- Reconocer la importancia de la Gestión de Configuración de Software.
- Conocer técnicas y herramientas para realizar pruebas y revisiones técnicas al software.
- Integrar por medio de casos prácticos concretos los conocimientos adquiridos en la parte teórica, empleando así las técnicas y herramientas de aplicación de la ingeniería de software.

Bloques de contenidos	Prácticas formativas Profesionalizantes
<p>Bloque: Disciplinas implicadas en el Desarrollo de Software</p> <ul style="list-style-type: none"> • Proceso de Desarrollo definidos y empíricos. • Procesos de Desarrollo y Modelos de Proceso. Criterios para elección de ciclos de vida en función de las necesidades del proyecto y las características del producto. • Disciplinas que conforman la Ingeniería de Software: disciplinas técnicas, disciplinas de gestión y de soporte. • Gestión de Configuración de Software • Testing de Software. • Revisiones técnicas al software. 	<ul style="list-style-type: none"> • Abordaje del concepto procesos para el desarrollo de software. • Aplicación de las disciplinas de ingeniería de software en un proyecto de desarrollo. • Análisis y selección de un ciclo de vida para un proyecto. • Abordaje del concepto de proyecto de desarrollo de software para contextualizar. • Uso de prácticas de Gestión de configuración de software • Aplicación de revisiones técnicas y testing del software.

<p>Bloque: Gestión de Proyectos</p> <ul style="list-style-type: none"> • Gestión de Proyectos de Desarrollo: tradicional y ágil. • Introducción a los Métodos Ágiles. • Manifiesto Ágil. • Requerimientos en ambientes ágiles - User Stories. • Métodos Ágiles: SCRUM, XP y TDD 	<p>A partir de situaciones problemáticas definidas por el equipo docente, los estudiantes participarán de dinámicas que posibiliten la vivencia de las prácticas propuestas por el marco de trabajo de gestión ágil de proyectos.</p>
---	---

Denominación de Módulo: **Programación Orientada a objetos**

Tipo de Módulo: Específico

Carga Horaria: **150** hs. reloj

Carga Horaria de Práctica Profesionalizante: **90 horas Reloj**

Presentación:

El módulo Programación orientada a Objetos tiene, como propósito general, contribuir a que los estudiantes desarrollen capacidades técnicas de programación con objetos. Profundiza y amplía las capacidades construidas en el módulo **Técnicas de Programación** dado que se emplean las herramientas adquiridas en este último, en una nueva modalidad de resolución de problemas.

Este módulo se constituye, así en un espacio de formación que permite a los estudiantes desarrollar saberes propios de la formación específica de la figura profesional de “Programador”.

En este contexto se entiende por *orientación a objetos* a un paradigma de programación que facilita la creación de software de calidad, debido a sus características específicas que potencian el mantenimiento, la extensión y la reutilización del software generado. Los programas se organizan como colecciones cooperativas de objetos, cada uno de los cuales representa una instancia de alguna clase, y cuyas clases son miembros de una jerarquía de clases vinculadas mediante relaciones de herencia.

Es de central importancia que las actividades de enseñanza de la programación orientada a objetos se asocien a prácticas cercanas (simuladas o reales) al tipo de intervención profesional del Programador.

Para la organización de la enseñanza de esta unidad curricular se han organizado los contenidos en tres bloques:

- **Fundamentos de la Programación Orientada a Objetos**
- **Metodología de desarrollo**
- **Lenguaje de POO**

El bloque de **Fundamentos de la Programación Orientada a Objetos** presenta los conceptos básicos de este paradigma: abstracción, encapsulamiento, modularización, jerarquía de clases y jerarquía de partes, polimorfismo y relaciones entre clases. Este bloque mantiene estrecha relación con el bloque **Lenguaje de POO**, ya que los conceptos que se abordan en este bloque, se implementan y desarrollan inmediatamente en el lenguaje seleccionado. Esta relación permite lograr una mayor comprensión de los conceptos y un acercamiento temprano al lenguaje y a los principios de desarrollo de software de calidad.

El bloque **Metodología de desarrollo** aborda las técnicas de resolución de problemas informáticos bajo la óptica del paradigma Orientado a Objetos utilizando un proceso de desarrollo y un lenguaje de modelado unificado (Proceso Unificado de Desarrollo / Lenguaje de Modelado Unificado).

El bloque **Lenguaje de POO** tiene, como núcleo central, la elaboración y la construcción de aplicaciones implementando los conceptos de POO y el modelado de situaciones problemáticas en un lenguaje adecuado al paradigma en un entorno de desarrollo corporativo o abierto.

La organización del programa curricular, que se presenta en este documento, sigue una secuencia que toma como punto de partida la identificación de las clases que modelan el problema, sus relaciones y representación mediante UML y por último su codificación en un lenguaje de programación orientado a objetos, dando como resultado el producto software.

El módulo “Programación orientada a objetos” recupera e integra conocimientos, saberes y habilidades cuyo propósito general es contribuir al desarrollo de los estudiantes de una formación especializada, integrando contenidos, desarrollando prácticas formativas y su vínculo con los problemas característicos de intervención y resolución técnica del Programador, en particular con **las funciones que ejerce el profesional en relación a:**

- **Interpretar especificaciones de diseño de las asignaciones a programar en el contexto del desarrollo de software en el que participa.**

Este módulo se orienta al desarrollo de las siguientes **capacidades profesionales referidas al perfil profesional en su conjunto:**

- Interpretar las especificaciones de requerimientos y diseño.
- Analizar el problema a resolver.
- Interpretar el material recibido y clarificar eventuales interpretaciones.
- Determinar el alcance del problema y convalidar su interpretación a fin de identificar aspectos faltantes.
- Comprender lo especificado observando reglas del lenguaje de POO.
- Comunicarse en un lenguaje preciso y adecuado con los integrantes del equipo de proyecto.

En relación a las **prácticas formativas de carácter profesionalizante**, son un eje estratégico de la propuesta pedagógica para el ámbito de la FP, al situar al participante en los ámbitos reales de trabajo con las problemáticas características que efectivamente surgen de la planificación de procedimientos o secuencias de actividades asociada al desarrollo de algoritmos y la resolución de problemas.

Para el caso del presente modulo las **prácticas formativas profesionalizantes** y los objetivos de aprendizajes se organizan para el desarrollo de:

- Práctica de resolución de una situación problemática, real o simulada de acuerdo a especificaciones de diseño, desarrollando aplicaciones que den solución a problemas específicos.

Bloques de contenidos	Prácticas Formativas Profesionalizantes
<p>Bloque: Programación Orientada a Objetos</p> <ul style="list-style-type: none"> • El modelo de objetos • Conceptos de <ul style="list-style-type: none"> • Clase y objeto • Atributos y métodos • Estado y comportamiento • Mensaje entre objetos. • Encapsulamiento de la información <ul style="list-style-type: none"> • Niveles de acceso • Tiempo de vida de los objetos. • Abstracción y modularización. • Herencia <ul style="list-style-type: none"> • Especialización vs generalización • Superclase y subclase • Clases abstractas • Polimorfismo y sobrecarga de operadores • Construcción de un diseño modular: Acoplamiento -Cohesión • Realización. Uso de interfaces <p>Bloque: Metodología de desarrollo de software</p> <ul style="list-style-type: none"> • Modelado con UML <ul style="list-style-type: none"> • Diagrama de Clases • Diagrama de Casos de Uso 	<p>A partir de situaciones problemáticas definidas por el equipo docente o a partir de un diseño existente, los estudiantes elaborarán programas informáticos que permitan resolver el requerimiento planteado, en lenguaje orientado a objetos. Mediante estas prácticas, se espera que las/os estudiantes puedan:</p> <p>Analizar y valorar la POO. Modelar objetos de acuerdo a los problemas a resolver</p> <p>Usar notación UML para construir modelar, implementando los mismos en el lenguaje de programación orientado a objetos elegido.</p> <p>Analizar y modelar requerimientos mediante diagramas de Casos de Uso.</p> <p>Construir vistas dinámicas del sistema con diagramas de Secuencia.</p> <p>Diseñar y programar software considerando criterios de calidad (alta cohesión y un bajo acoplamiento entre las clases).</p> <p>Documentar clases, métodos y secciones de código.</p> <p>Incluir código para la detección y manipulación de errores en tiempo de ejecución.</p> <p>Diseñar y construir una interacción para el usuario adecuada a sus expectativas y necesidades, aprovechando las posibilidades del lenguaje.</p>

<ul style="list-style-type: none"> • Diagrama de Secuencia • Diagrama de Máquina de Estados <p>Bloque: Lenguaje de Programación Orientada a Objetos.</p> <ul style="list-style-type: none"> • Sintaxis y estructura del lenguaje • Estructura de una clase <ul style="list-style-type: none"> • Variables de instancia • Métodos de acceso y de modificación • Constructores • Constructores y métodos sobrecargados. • Métodos accesorios o auxiliares • Documentación de clases y de métodos • Constantes y variables de clase. • Interfaz e implementación de una clase • Librería de clases del lenguaje <ul style="list-style-type: none"> • Clases de fines específicos • Agrupamiento de objetos. <ul style="list-style-type: none"> • Colecciones de tamaño fijo y de tamaño variable. • Manejo de errores. • Noción de evento • Programación por interfaces • Diseño de interacción de usuario 	
---	--

Denominación de Módulo: **Base de Datos**

Tipo de Módulo: Específico

Carga Horaria: **70** hs. reloj

Carga Horaria de Práctica Profesionalizante: **50 horas Reloj**

Presentación:

El módulo Base de datos tiene, como propósito general, que los estudiantes construyan los conceptos y desarrollen las técnicas de manejo y creación de consulta y manipulación de bases de datos.

Para la organización de la enseñanza de esta unidad curricular se han organizado los contenidos en dos bloques:

- **Organización lógica de los datos**
- **Manipulación de base de datos**

El bloque **Organización lógica de los datos** aborda, como primera aproximación, la estructura y organización de los datos en bases de datos de tipo relacionales, lo cual implica el reconocimiento de los componentes que la estructuran, la vinculación y optimización de los datos para, posteriormente manipular la base de datos, mediante un lenguaje de consulta.

El bloque **Manipulación de base de datos** tiene como núcleo central, el acceso a los datos mediante un lenguaje o Sistema Gestor (libre o propietario) brindando los conocimientos esenciales vinculados con la organización de los datos en relación con su almacenamiento.

El propósito formativo de este módulo, es que los estudiantes construyan habilidades y conocimientos para elaborar, normalizar, consultar y manipular bases de datos relacionales en base a especificaciones obtenidas.

El módulo “Base de datos” recupera e integra conocimientos, saberes y habilidades cuyo propósito general es contribuir al desarrollo de los estudiantes de una formación especializada, integrando contenidos, desarrollando prácticas formativas y su vínculo con los problemas característicos de intervención y resolución técnica del Programador, en particular con **las funciones que ejerce el profesional en relación al:**

- **Manejo y manipulación de los datos y su relación con las aplicaciones desarrolladas o a desarrollar**

Este módulo se orienta al desarrollo de las siguientes **capacidades profesionales referidas al perfil profesional en su conjunto:**

- Analizar los datos necesarios para dar soporte a la aplicación
- Comprender lo especificado utilizando las reglas del lenguaje de consulta

En relación a las **prácticas formativas de carácter profesionalizante**, se definen como uno de los ejes estratégicos de la propuesta pedagógica para el ámbito de la FP, el situar al participante en los ámbitos reales de trabajo, con las problemáticas características que efectivamente surgen de la planificación de procedimientos o secuencias de actividades asociada al uso y manipulación de los datos, del lenguaje de consulta y la resolución de problemas relativos al flujo de datos dentro de los problemas de tipo computacional.

Para el caso del presente modulo las **prácticas formativas profesionalizantes** y los objetivos de aprendizajes se organizan para el desarrollo de:

- Práctica de resolución de una situación problemática, real o simulada de acuerdo a especificaciones de diseño, utilizando un lenguaje de consulta y manejo de datos necesarios para dicha situación.

Contenidos	Prácticas Formativas Profesionalizantes
<p>Bloque: Organización lógica de los datos</p> <ul style="list-style-type: none"> • Concepto de datos, procesos, salida. • Noción de registro y Concepto de archivo. • Noción de base de datos. • Ventajas de las Bases de Datos. • Concepto de Tabla. • Concepto de Entidad. • Relaciones entre entidades. • Atributo de las entidades. • Clave principal y Clave secundaria. • Cardinalidad de las relaciones. • Diagrama Entidad-Relación (ER). • Concepto de normalización y redundancia. • Consultas anidadas. • Tablas relacionadas. • Consultas de datos. • Altas, Bajas y Modificaciones (ABM). <p>Bloque: Manipulación de Base de datos. Lenguaje de Consulta</p> <ul style="list-style-type: none"> • Software libre. • Software propietario. <ul style="list-style-type: none"> • Construcción de Base de datos. • Consultas sobre una tabla • Selección simple. • Eliminación de respuestas duplicadas. • Selección ordenada. 	<p>A partir de situaciones problemáticas definidas por el equipo docente, aplicaciones informáticas existentes y/o desarrolladas por ellos, las/os estudiantes analizarán y vincularán los datos a los programas para resolver el requerimiento planteado, elaborando las consultas correspondientes en el sistema gestor adecuado. Mediante estas prácticas, se espera que las/os estudiantes puedan:</p> <p>Identificar las ventajas del uso de las bases de datos vinculadas con la consulta y la actualización de los datos</p> <p>Diferenciar los tipos de datos según las operaciones que se deban realizar sobre ellos.</p> <p>Construir los diagramas Entidad-Relación relacionadas con las situaciones problemáticas planteadas.</p> <p>Resolver problemas de redundancia en los datos</p>

<ul style="list-style-type: none"> • Selección condicionada. • Operadores de comparación • Operadores lógicos. • Campos calculados. • Funciones agregadas de dominio (mínimo, máximo, suma, promedio). • Selección con agrupamientos (agrupar). • Altas, bajas y modificaciones (ABM). • Actualización de un registro. • Actualización de registros múltiples • Actualización condicionada. • Inserción y eliminación de un registro. • Eliminación de múltiples registros. • Uso de Transacciones 	
---	--

IX. Referencial de ingreso

Haber completado el nivel de la Educación Secundaria, acreditable a través de certificaciones oficiales del Sistema Educativo Nacional (Ley N° 26.206) y conocimiento de habilidades básicas en la operación informática.
