

DISEÑO CURRICULAR

TÉCNICO EN PROGRAMACIÓN

ANEXO

TÉCNICO EN PROGRAMACIÓN

1. Identificación del título

- 1.1. Sector/es de actividad socio productiva: Informática 1 (Software y servicios informáticos)
- 1.2. Denominación del perfil profesional: Técnico en Programación
- 1.3. Familia profesional: Informática
- 1.4. Denominación del título de referencia: Técnico en Programación
- 1.5. Nivel y ámbito de la trayectoria formativa: nivel Secundario y ámbito de la Educación Técnica de la modalidad de Educación Técnico profesional.

2. Referencial al Perfil Profesional

El perfil profesional del Técnico en Programación alude al conjunto de realizaciones profesionales que el técnico puede demostrar en las diversas situaciones de trabajo propias de su área ocupacional, una vez que ha completado el proceso formativo.

Este perfil involucra un conjunto de competencias que asegura un mayor nivel de especificidad y profundización en ámbitos contextualizados del saber, saber hacer y saber ser, dentro del sector profesional de la Informática. Se articula en torno a núcleos curriculares comunes y orientados, a partir de las demandas socio-productivas del sector y la realidad del medio industrial y de servicios.

2.1. Alcance del Perfil Profesional

El Técnico en Programación estará capacitado para realizar programas o componentes de sistemas de computación – interpretar especificaciones de diseño, documentar los productos realizados, verificar los componentes programados, buscar causas de malfuncionamiento y corregir los programas o adaptarlos a cambios en las especificaciones – desarrollando las actividades descritas en el perfil profesional y cumpliendo con los criterios de realización establecidos para las mismas en el marco de un equipo de trabajo organizado por proyecto.

Este Técnico en Programación participa en proyectos de desarrollo de software desempeñando roles que tienen por objeto producir programas, módulos o componentes de sistemas de computación. Estos módulos suelen integrarse en aplicaciones que interactúan con otras ya existentes desarrolladas con la misma o diferente tecnología.

Sus actividades profesionales cubren las siguientes áreas:

“Interpretar especificaciones de diseño o requisitos de las asignaciones a programar”, en el contexto del proyecto. Convalida su propia interpretación con quienes la hayan realizado o provisto.

“Planificar su trabajo en el contexto del equipo de desarrollo del proyecto y de la tecnología a utilizar”.

Identifica aspectos de posible dificultad o riesgo, evalúa las características del entorno, tiempos y esfuerzos para lograr la solución del problema, considera la posibilidad de subdividir la asignación en pasos o componentes menores y establece un orden o secuencia de trabajo.

Analiza estrategias para desarrollar la asignación recibida e investiga para refinar aspectos de diseño, algoritmos o estructuras de datos, busca componentes disponibles y adecuados y plantea soluciones alternativas para ser evaluadas en el contexto del proyecto.

“Producir programas, módulos o componentes de sistemas de computación”, respondiendo a necesidades específicas en el contexto de la tecnología a utilizar. Para ello desarrolla algoritmos que den solución a los problemas a resolver y define estructuras de datos adecuadas a los mismos. También construye o modifica clases y objetos, reutiliza componentes existentes o diseña interfaces. Modifica códigos ya escritos para corregir errores o para cambiar funcionalidades o comportamientos de productos existentes.

Integra los componentes programados en aplicaciones que interactúan con otras ya existentes desarrollados con las mismas o diferentes tecnologías.

“Verificar y depurar el producto desarrollado”, para asegurarse que cumple con las especificaciones recibidas. Implica la implementación de un conjunto de pruebas para detectar comportamientos o resultados no previstos y buscar sus causas. Comprende revisar códigos para encontrar las partes o instrucciones que provocan los malfuncionamientos y definir las acciones correctivas.

“Realizar revisiones cruzadas de código o de interfaces”, con otros programadores o con especialistas, para evaluar el uso eficiente de recursos y del ambiente de desarrollo, y aporten observaciones con propuestas de cambio con el objeto de mejorar la calidad, mantenibilidad y eficiencia del producto.

“Realizar la documentación técnica y de usuarios de acuerdo con los requerimientos funcionales y técnicos de las aplicaciones y sistemas”.

Documenta su trabajo para que resulte interpretable y utilizable por otros. Esto incluye comentar el código, complementar documentos de diseño, confeccionar o completar reportes de incidentes, adjuntar resultados de pruebas o advertencias sobre posibles limitaciones de la solución. También incluye la identificación de las versiones producidas.

“Explotar las funcionalidades de los sistemas de información, hardware, software y redes”, para la realización de las actividades. Implica conocer y saber utilizar eficientemente recursos de hardware, software y redes para utilizar los ambientes que necesite para el desarrollo su trabajo.

La actividad del programador es no rutinaria a pesar de que muchas veces se reutilicen partes ya existentes. Cada asignación representa la necesidad de dar satisfacción a determinados requisitos. Ello requiere comprender el problema y la arquitectura en la que estará inserta la solución, idear estrategias de resolución y ser capaz de aplicar debidamente el lenguaje y

IF-2018-31697359-GDEBA-DETECDGCYE

ambiente de programación a emplear, así como aplicar buenas prácticas de programación, lo que incluye documentar decisiones significativas de diseño y las limitaciones que tendrá el artefacto construido.

Para poder desarrollar plenamente su profesionalidad, el técnico tiene que poseer ciertas capacidades que resultan transversales a todas sus funciones y tienen que ser desarrolladas durante el transcurso de su formación. Estas son:

Abstracción - Implica descartar o reducir detalles poco significativos de la información sobre un problema para concentrarse en pocos elementos por vez, lo que resulta en una reducción de la complejidad que permita conceptualizar de modo más simple un dominio de problemas para facilitar su comprensión y manejo en forma genérica de sus posibles soluciones.

Pensamiento combinatorio - Conduce a la consideración sistemática de un conjunto de alternativas, lo que incluye el manejo mental de muchas variables o detalles del problema sin perder nunca de vista el concepto o la estrategia general de resolución.

Autorregulación - Implica manejarse respetando reglas y limitaciones, tanto explícitas como implícitas, sean éstas propias o del grupo de trabajo; actuar ateniéndose a un orden propio que le facilite el acceso a lo que puede necesitar, reconocer y guardar; referenciar la información y registrarla de tal manera que le facilite acceder posteriormente en forma rápida para evaluarla y recuperarla.

Comunicación apropiada - Implica una disposición a reconocer que existen otros que pueden aportar información útil o a quienes puede interesarle lo que hace. Supone reconocer su rol y el de cada integrante del proyecto, transmitir la información necesaria en forma precisa y en un lenguaje apropiado para el entendimiento mutuo en interacciones individuales o grupales, o en forma escrita, utilizando, si es necesario para ello, el idioma inglés, que debe interpretar con propiedad a nivel técnico.

Trabajo en equipo - Implica adoptar una actitud abierta, estar dispuesto a compartir información y conocimientos, a tomar en cuenta a los usuarios del producto que está construyendo, a brindar, pedir y aceptar ayuda cuando ésta resulte necesaria para facilitar su propia labor o la de otro integrante del equipo. Comprende al equipo del proyecto, incluyendo a los usuarios que participan del mismo.

2.2. Funciones que ejerce el profesional

A continuación se presentan funciones y sub-funciones del perfil profesional del técnico de las cuales se pueden identificar las actividades profesionales:

Interpretar especificaciones de diseño o requisitos de las asignaciones a programar

Esto comprende:

- *Analizar el problema a resolver.*
- *Determinar el alcance del problema.*
- *Validar la coherencia e integridad de las especificaciones.*
- *Convalidar su propia interpretación con quienes lo hayan realizado o provisto.*

El Técnico en Programación de Computadores analiza el problema a resolver, que puede estar especificado formal o informalmente como instrucciones de diseño o requerimientos del usuario. Para ello resulta necesario interpretar críticamente el material recibido y validar si

IF-2018-31697339-GDEBA-DETECDGCYE

todo lo pedido resulta coherente entre sí o con otros aspectos que conozca del proyecto para clarificar eventuales malas interpretaciones o desacuerdos y convalidar su interpretación con el responsable del proyecto.

Esto implica que tiene que ser analítico y poseer una buena capacidad de abstracción para ser capaz de comprender lo especificado, observando reglas de los lenguajes en que está expresado (*storyboards*, casos de uso, *UML*, otros tipos de diagramas, diccionarios de datos), describir en sus propios términos el problema, identificar puntos ambiguos, aspectos faltantes o eventuales contradicciones entre distintos requisitos a cumplir o incoherencias entre estos y otros aspectos conocidos del proyecto. También debe ser capaz de comunicarse en un lenguaje preciso y adecuado con el líder o usuario con quien discuta su interpretación para convalidarla.

Planificar su trabajo y analizar estrategias para desarrollar la asignación recibida

Esto comprende:

- *Identificar aspectos críticos.*
- *Dividir la asignación en subtarefas o productos intermedios.*
- *Establecer un orden o secuencia de trabajo.*
- *Estimar tiempos de realización.*
- *Establecer prioridades y necesidades de apoyo y consulta para refinar aspectos ambiguos o insuficientemente conocidos del diseño.*
- *Utilizar metodologías de búsqueda de información de fuentes confiables.*
- *Aplicar técnicas y metodologías para la resolución de problemas.*

Para realizar esto el técnico tiene que contemplar requerimientos técnicos y funcionales, a cubrir, estimar dificultades y tiempos, imaginar y desarrollar alternativas de solución a fin de organizar su tarea y prever sus tiempos y posibles dificultades.

Esto implica que tiene que ser capaz de averiguar y completar detalles de diseño, considerar si existen bibliotecas con patrones, clases, rutinas o módulos que pueda utilizar, eventualmente construir prototipos y demos para visualizar la propuesta y comparar ventajas y desventajas de las distintas alternativas para seleccionar la que considera más adecuada para planificar su tarea, anticipando posibles riesgos a enfrentar en su asignación para solicitar la colaboración o asesoramiento que corresponda.

Al hacer esto utiliza su experiencia acumulada, consulta bibliotecas o listas de discusión en Internet y arma su propio repertorio de material a utilizar.

Producir programas, módulos o componentes de sistemas de computación en el contexto de la tecnología a utilizar

Esto comprende:

- *Desarrollar algoritmos que den solución a los problemas asignados.*
- *Definir el código.*
- *Definir estructuras de datos eficaces y explotarlos con eficiencia.*
- *Definir, desarrollar instancias y completar clases y objetos apropiados para representar el problema a resolver.*
- *Diseñar interfaces respetando el estilo del usuario y del contexto previsto.*

Para realizar esto el técnico utiliza patrones, reutiliza código existente adaptándolo o complementándolo a su nueva función o redacta código nuevo aplicando sus conocimientos de programación, respetando buenas prácticas y las normas establecidas para asegurar la calidad del proyecto. Esto implica el dominio del lenguaje y ambiente de desarrollo utilizados en el proyecto, así como la tecnología en la cual va a ser implementada la solución, así como la aplicación de criterios de simplicidad y coherencia en la elaboración de interfases.

Verificar el producto desarrollado

Esto comprende:

- *Analizar y registrar todos los procesos alternativos importantes.*
- *Procesar el producto obteniendo y registrando los resultados.*

Para realizar esto el técnico determina las necesidades de cobertura en función de las características de su asignación y normas establecidas para asegurar la calidad del proyecto, identifica las clases de equivalencia de datos utilizados internamente o intercambiados y diseña los casos de prueba, tomando en cuenta la estructura del artefacto y las condiciones de borde, así como prepara el entorno de pruebas, incluyendo los scripts y datos necesarios. Esto implica el dominio de conceptos de testing y de herramientas utilizadas para establecer el ambiente de testing unitario. Realiza las pruebas correspondientes, registrando los datos y resultados alcanzados, así como las acciones correctivas realizadas para solucionar las fallas encontradas.

Depurar estructuras lógicas o códigos de programas

Esto comprende:

- *Relacionar resultados insatisfactorios con los datos o porciones de código que los originaron.*
- *Analizar estos datos y/o partes del código que causaron el mal funcionamiento y determinar el tipo de corrección o reemplazo.*
- *Verificar que la corrección y/o reemplazo solucionen el mal funcionamiento.*

Para realizar esto el técnico tiene que relacionar resultados insatisfactorios con probables causas y recorrer la estructura y código del programa para identificar el origen del error en el código que origina el mal funcionamiento. Una vez identificado el error, corresponde razonar sobre el tipo de corrección o reemplazo y analizar que el nuevo código no introduzca otros problemas.

Esta actividad se aplica tanto a programas propios como ajenos, que agregan un nivel de dificultad al no tenerse presente su estructura o no conocerse el estilo del código. También consulta a pares y al líder del equipo de trabajo para reflexionar y recibir ayuda que le permita resolver problemas encontrados o aporta sus conocimientos y capacidad de reflexión a otros, y participa de foros y listas temáticas para encontrar soluciones o elementos reutilizables.

Realizar revisiones cruzadas de código o de interfaces

Esto comprende:

- *Revisar el cumplimiento de estándares y de especificaciones.*
- *Revisar las interfaces desarrolladas con otros programadores o con especialistas para evaluar el uso eficiente de recursos y del ambiente.*

IF-2018-31697559-GDEBA-DETECDGCYE

- *Reportar observaciones sobre propuestas de cambio.*

Para realizar esto el técnico revisa con otros programadores o especialistas si las interfaces resultan coherentes dentro del estilo del sistema, amigables para el usuario y para personas con capacidades diferentes; que los códigos producidos no demanden tiempos de proceso, asignaciones de memoria o almacenamiento excesivos para el contexto.

Esto implica la capacidad de reconocer estructuras y un dominio del lenguaje de programación, así como el conocimiento de buenas prácticas de programación y normas de documentación.

También la capacidad de trabajar en equipo y de comunicación para informar las observaciones recibidas y presentar propuestas de cambio significativas en forma verbal o escrita.

Realizar la documentación técnica y de usuarios de acuerdo con los requerimientos funcionales y técnicos de las aplicaciones y sistemas.

Esto comprende:

- *Describir características, relaciones y limitaciones de nuevas clases utilizando diagramas u otros elementos.*
- *Intercalar en el código descripciones de sus características y limitaciones.*
- *Registrar decisiones de diseño, elementos utilizados y resultados de pruebas.*
- *Plasmar incidentes, errores, soluciones y tiempos utilizados.*
- *Identificar cada versión del producto de acuerdo a estándares.*

El técnico realizará la documentación con claridad, consistencia y completitud. Describe que hace cada parte del código y por qué se incluye, datos, otros elementos o situación que lo originaron; registros y evidencias de las actividades realizadas y de los incidentes observados, identifica cada versión de acuerdo a estándares.

Para lograr un desempeño competente en sus actividades profesionales, el desarrollador de software, además de realizar las actividades previstas en su perfil profesional e incluidas aquí en la descripción de las funciones que realiza, tiene que conocer ciertos aspectos de la tecnología de la información que le sirven de base para poder desarrollar competentemente sus funciones profesionales. Al dominio de estos aspectos lo hemos denominado:

Desempeño de base – Esto implica conocer y saber utilizar con propiedad y en condiciones de seguridad recursos de hardware, software y redes para emplear los ambientes que necesite para el desarrollo y la verificación del software, mantener los repositorios de información que necesite utilizar y disponer de los productos de su trabajo en condiciones de confiabilidad.

Esto comprende:

- *Configurar lógicamente el sistema al entorno de trabajo para desarrollar y probar los programas.*
- *Organizar y mantener componentes de software y datos de prueba en sistemas de archivos, utilizando las utilidades comunes al proyecto*
- *Recuperar, presentar y distribuir información en su estación de trabajo o a través de la red.*
- *Respetar procedimientos propios o de la organización que aseguren la integridad, disponibilidad y seguridad del sistema y de la información durante el desarrollo y verificación de programas.*

IF-2018-31697559-GDEBA-DETECDGCYE

Integrar la producción propia en el conjunto del proyecto identificándolas de acuerdo a los procedimientos de administración de versiones en uso por el proyecto.

Para realizar esto, el técnico tiene que poseer un dominio de la tecnología, tanto de hardware y redes, como de software de base, así como una disciplina de trabajo que le permita organizar y administrar sus propias herramientas y repositorios de información sin afectar a las actividades de otros y entregar los productos de su labor correctamente identificados de acuerdo a lo establecido para el proyecto, manteniendo un adecuado seguimiento de su labor que permita responsabilizarse por lo realizado.

2.3 Área Ocupacional

Este técnico se ocupa en organizaciones de diversos tipos que tengan que desarrollar software. Empresas que realizan desarrollo de software por encargo de organizaciones locales o extranjeras, que proveen software junto con otros servicios de asesoramiento y consultoría, y, en menor número, 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 especificaciones de requerimientos, ya sean estas 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 elaboradas por algún analista funcional integrante del equipo de trabajo del proyecto. El equipo de desarrollo suele estar encabezado por un gerente o líder responsables por el proyecto e integra diversos roles ocupacionales, como el de un arquitecto de software, que establece el diseño general del sistema y especificaciones de calidad de la solución, una serie de programadores, que son quienes lo construyen y un grupo de testing, que son los encargados de verificar que el software producido cumpla los requisitos, tanto funcionales como de comportamiento, oportunamente establecidos. 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 desarrolladores. Eventualmente pueden participar diseñadores gráficos y especialistas en otros aspectos específicos.

La posición ocupacional de este técnico suele denominarse analista-programador o programador, aunque últimamente se está generalizando una denominación más abarcativa y menos categorizante de desarrollador de software. Integra equipos de proyecto dedicados al desarrollo o mantenimiento de software y recibe asignaciones específicas que tiene que resolver en lapsos que suelen medirse en términos de días o semanas, produciendo artefactos que satisfagan especificaciones y se integren al sistema objeto del proyecto.

A partir de especificaciones de diseño y del conocimiento de la arquitectura del sistema, los programadores (también denominados analistas programadores o simplemente desarrolladores) completan el diseño en detalle de la parte que les fuera asignada, la construyen, preferiblemente en base a artefactos de software ya existentes y adaptando o escribiendo lo que sea necesario, así como documentándola para facilitar su testeo y posterior mantenimiento por otros, verifican unitariamente lo producido y lo entregan para ser probado integralmente e integrado al resto. Habitualmente, los desarrolladores, que pueden estar

IF-2018-31697539-GDEBA-DETECDGCYE

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 y los que tienen mayor experiencia suelen brindar orientación (coaching) a los más noveles.

Resuelve estas asignaciones individualmente o trabajando en pares, recibiendo la supervisión y asesoramiento de un líder de proyecto o de grupo, con quien consulta dudas y decisiones significativas o comunica inconvenientes. También recibe apoyo y brinda colaboración a otros miembros del grupo. Asimismo, 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. Por otra parte, Técnicos en Programación o profesionales equivalentes con capacidad emprendedora pueden y suelen asociarse entre ellos para generar sus propias empresas para brindar servicios de desarrollo y proveer software a terceros.

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

- Servicios informáticos para pequeñas y medianas empresas en áreas de análisis y programación 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 implantación y 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 y de redes de área local.
- Asesoramiento técnico y venta de sistemas y aplicaciones informáticas.

2.4 Habilitaciones profesionales

Las actividades que realiza y para las cuales está capacitado el Técnico en Programación, así como el ámbito de su desempeño y el campo y condiciones de su ejercicio profesional son los descriptos en el Perfil Profesional correspondiente.

Si bien las actividades de este técnico no están orientadas a un tipo de software en particular, conviene tomar en cuenta que el software es utilizado crecientemente en sistemas que afectan a la seguridad pública. Estos sistemas, denominados críticos para la seguridad, son lo que, en un sentido general, involucran riesgos que conllevan la posibilidad de pérdidas inaceptables (daños para la salud o aún la vida humana, daños a la propiedad, contaminación ambiental, conflictos sociales, grandes pérdidas monetarias).

En función de estos riesgos, se establecen las siguientes habilitaciones profesionales, para el Técnico en Programación, con las limitaciones o exclusiones que se indican en cada caso.

Estas habilitaciones tienen efecto para su desempeño en forma autónoma o asumiendo plenamente la responsabilidad por los resultados que obtenga su grupo de trabajo.

IF-2018-31697559-GDEBA-DETECDGCYE

- Desarrollar y mantener programas de software de complejidad media, correspondiente a sistemas de información o vinculado indirectamente al hardware o a sistemas de comunicación de datos, respondiendo a especificaciones.

Queda excluido de esta habilitación el software correspondiente a sistemas críticos para la seguridad, como es el caso de los que involucren el procesamiento de información que conlleve riesgos efectivos para terceros. Particularmente, queda excluido el software destinado a: control de equipos y procesos médicos, industriales o de domótica que puedan poner en riesgo inmediato o mediato la salud de personas; procesamiento de información crítica para los individuos, como ser la que sirva para corroborar su identidad o características de su estado de salud, para demostrar situaciones legal, fiscal, patrimonial u otras que afecten a su patrimonio o a sus libertades; procesamiento en línea de transacciones financieras importantes.

En estos casos, requerirá la supervisión de profesionales habilitados.

- Operar actividades de testing de software de aplicaciones
- Redactar documentación técnica.

3. En relación con la trayectoria formativa

Los planes de estudio a ser presentados para su homologación deberán evidenciar el trayecto formativo completo que conduce a la emisión del título técnico de nivel medio, independientemente de la organización institucional y curricular adoptada, de manera tal que permitan identificar los distintos tipos de contenidos a los que hace referencia.

Deberán identificarse los campos de formación general, de formación científico-tecnológica, de formación técnica específica y de prácticas profesionalizantes.

De la totalidad de la trayectoria formativa y a los fines de homologar títulos de un mismo sector profesional y sus correspondientes ofertas formativas, que operan sobre una misma dimensión de ejercicio profesional, se prestará especial atención a los campos de formación científico-tecnológica, de formación técnica específica y de prácticas profesionalizantes. Cabe destacar que estos contenidos son necesarios e indispensables pero no suficientes para la formación integral.

3.1 Formación general

El campo de la formación general es el que se requiere para participar activa, reflexiva y críticamente en los diversos ámbitos de la vida social, política, cultural y económica y para el desarrollo de una actitud ética respecto del continuo cambio tecnológico y social. Da cuenta de las áreas disciplinares que conforman la formación común exigida a todos los estudiantes del nivel medio, de carácter propedéutica. A los fines del proceso de homologación, este campo, identificable en el plan de estudios a homologar, se considerará para la carga horaria de la formación integral del técnico.

3.2 Formación científico-tecnológica

El campo de la formación científico-tecnológica identifica los conocimientos, habilidades, destrezas, valores y actitudes que otorgan particular sostén a las técnicas y métodos de trabajo.

IF-2018-31697539-GDEBA-DETECDGCYE

así como las tecnologías propias del campo profesional en que se desempeña este técnico. Estos saberes, que sustentan a la formación específica, profundizan o complementan a los de la formación general, por lo que conviene prever una adecuada articulación con los mismos, así como también resulta importante que el diseño curricular tome en cuenta su relación con los aspectos técnico específico. Las áreas disciplinares relacionadas con la formación científico tecnológica de la trayectoria formativa de este técnico son:

Provenientes del campo de la matemática y la lógica

La matemática y la lógica desarrollan capacidad de pensamiento abstracto, razonamiento por inferencias y análisis combinatorio de alternativas requerido por el pensamiento computacional.

Además, la resolución de asignaciones de programación requiere permanentemente la necesidad de resolver problemas, a lo cual contribuyen las estrategias desarrolladas para la solución de problemas matemáticos. En consecuencia, se recomienda adoptar un enfoque práctico, basado en el planteo de problemas, para abordar los siguientes contenidos: Conjuntos. Lenguaje coloquial, simbólico y gráfico; diagramas de Venn. Cardinalidad y numerabilidad. Operaciones con conjuntos: intersección, unión, diferencia, diferencia simétrica, complementación. Leyes de De Morgan. Particiones.

Sistemas de numeración. Concepto de número, formas de representación. Sistemas posicionales: binario, octal, hexadecimal. Operaciones en los distintos sistemas. Concepto de overflow.

Números reales: propiedades, operaciones, aproximación decimal, cálculo aproximado, redondeo y truncamiento y su influencia en los errores de cálculo, error absoluto y relativo.

Codificación de información. Sistemas de representación, operaciones aritméticas en punto flotantes, concepto de excepción.

Lógica Simbólica: Propositiones. Conectivos lógicos. Operaciones lógicas. Equivalencia lógica. Clasificación de proposiciones según tabla de verdad: tautologías, contradicciones. Reducción de expresiones lógicas a su mínima expresión.

Lógica de circuitos digitales. Expresiones lógicas y funciones booleanas. Nociones de lógica difusa.

Demostración matemática: demostración directa, por contraejemplo, principio de inducción completa. Definiciones matemáticas recursivas. Buen ordenamiento.

Vectores y matrices, producto escalar y cartesiano, operaciones con matrices, transposición de matrices. Determinantes. Sistemas de ecuaciones lineales, métodos de resolución directos y aproximados, incompatibilidad e indeterminación. Inecuaciones.

Funciones, concepto y representación, tipos de funciones y relaciones. Funciones elementales más comunes, operaciones con funciones elementales, funciones polinómicas (operaciones con polinomios, raíces), valor absoluto, potencial, exponencial, logarítmica y trigonométricas.

Representación de problemas por medio de funciones, métodos de solución de ecuaciones.

Probabilidades en espacios discretos: experimentos aleatorios, espacios muestrales, sucesos, probabilidad condicional e independencia. Combinatoria.. Variables aleatorias, distribuciones de probabilidad, esperanza matemática, varianza, ley de los grandes números. Nociones de estadística descriptiva, medidas de posición y dispersión, estimadores. Concepto de distribución de variable continua, distribuciones más comunes.

IF-2018-31697559-GDEBA-DETECDGCYE

Provenientes del campo de la física

La física facilita una comprensión razonada de los objetos del mundo real y el desarrollo de modelos abstractos que representen su comportamiento. También resulta útil el conocimiento de los movimientos para la programación de animaciones. En tal sentido, además de lo que establezca el currículo general del secundario, se contempla profundizar los siguientes contenidos:

Cinemática, dinámica, conceptos de inercia y estabilidad. Rozamiento y choque elástico. Cargas atractivas y repulsivas.

Por otra parte, se descuenta que los aspectos de electricidad, magnetismo, ondas, óptica y movimientos de rotación recibirán la atención necesaria en el currículo como para facilitar la comprensión de los fenómenos y dispositivos propios de las tecnologías de la información y las comunicaciones.

Provenientes de la campo de la computación

Este campo figura tanto en la formación científico-tecnológica como en la técnico-específica.

En la primera han sido previstos contenidos elementales y fundamentales de la disciplina de la computación que son de carácter general y deberían formar parte de la educación de cualquier ciudadano preparado para desenvolverse con soltura en la sociedad de la información. Estos contenidos pueden servir de motivación para quienes aspiren a desempeñarse en funciones técnicas del sector informático y conviene que se incorporen tempranamente a su formación. A la formación técnico-específica han sido asignados los que el Técnico en Programación va a poner en juego en forma directa al realizar sus actividades.

Introducción al mundo de la computación y al pensamiento computacional. La programación como forma de expresión.

Concepto de algoritmo, los pasos básicos en la resolución algorítmica de problemas (exploración y formulación del problema, examen de una muestra de casos particulares, estrategias de diseño, realización, prueba y verificación). El problema de la complejidad.

Estructuras fundamentales, variables, tipos, expresiones y asignaciones, entrada/salida, estructuras de control condicionales e iterativas, funciones y pasaje de parámetros, descomposición estructurada.

Concepto de lenguaje de alto nivel y la necesidad de traducción, comparación entre compiladores e intérpretes, aspectos de la traducción dependientes y no dependientes de la máquina. Programas generadores de código.

Como parte de la forma de adquirir estos aprendizajes y demostración práctica de los resultados alcanzados, los estudiantes tienen que utilizar lenguajes sencillos de programación que motiven el abordaje de problemas de interés para los alumnos y la práctica de pensamiento computacional. También se puede prever la integración de imágenes y sonido en estas realizaciones.

Provenientes de la tecnología de la información

Conceptos de tecnología de la información, evolución histórica, tecnologías que la integran, disciplinas que forman parte (ciencia de la computación, ingeniería de software, sistemas de información, ingeniería en computación) o aportan a la misma.

Evolución del computador, su organización y unidades funcionales que lo componen.

Arquitectura interna de computadores, unidad central de procesamiento, instrucciones y flujo

IF-2018-31697359-GDEBA-DEPECDCGYE

de la información Tipos y niveles de organización de la memoria interna y externa (sistemas de memoria, tecnologías y jerarquías, memoria caché, memoria virtual, dispositivos de almacenamiento secundario). Periféricos, clasificación y utilización; dispositivos de almacenamiento externo (discos duros, CDs, DVDs, tarjetas de memoria, otros). Estado actual de la tecnología. Procesadores multinúcleos, threading, computación paralela.

Funcionamiento de las instrucciones de un programa a nivel de una máquina simplificada (principalmente como medio de comprender características de su funcionamiento).

Provenientes de las organizaciones y los sistemas de información

Elementos de teoría general de los sistemas; objetivos, recursos, componentes, frontera, medio ambiente; niveles: suprasistema, sistemas pares, subsistemas; estructura, clasificación y características; enfoque sistémico de la organización. La información como recurso de las organizaciones y en el proceso de toma de decisiones: clases de decisión, proceso de toma de decisiones, características de las decisiones según niveles jerárquicos en la organización.

Elementos de estructura y comportamiento de las organizaciones, los recursos que administran las organizaciones, organización estructurada por funciones o líneas de productos, el manejo de sedes.

Concepto de proceso. Procesos del ciclo de ventas y cobranzas; del ciclo de compras y pagos.

Nociones de procesos de gestión y transformación de materiales y su organización.

Comprobantes usuales, requerimientos legales y fiscales. Concepto de recurso y su gestión.

Modelo hidráulico del movimiento y acumulación de bienes de cambio y dinero. El papel de los sistemas de información en la organización. Nociones de control interno. La contabilidad como sistema de información. Algunas características de organizaciones y procesos de servicios.

Los niveles de la administración: la planificación estratégica, el control de gestión, el control operativo y el tipo de sistemas de información asociados a los mismos.

Sistemas de información, métodos de procesamiento de datos; características, clasificación y función de la información. Función de un sistema de información. Sistemas de información típicos y aplicaciones usuales vinculados con la comercialización y distribución de bienes y servicios.

Provenientes del campo de la ética y del derecho

Importancia social y económica de los servicios de tecnología de la información, significado de Internet, valor de la información almacenada para las organizaciones, seguridad. Valor de la información para los individuos, normativa relativa a privacidad y “habeas data”. Bases de datos públicas y privadas. Propiedad de datos empresarios. Secretos comerciales e industriales. Contexto normativo: responsabilidades empresarias, contratos, responsabilidades del trabajador, leyes de protección de datos personales, propiedad intelectual del software y de contenidos, conceptos jurídicos aplicables a delitos informáticos.

Privacidad de datos personales. Normas que rigen el correo electrónico. Protección legal de la propiedad intelectual (incluyendo software), derechos de reproducción y derechos sobre marcas y patentes. Licencias de fabricación, de uso, GNU y “creative commons”.

Provenientes del campo del idioma inglés

Inglés técnico. Lectura e interpretación de textos e información técnica en inglés. Comprensión y producción de textos de complejidad creciente en inglés para comunicarse solicitando o aportando información técnica por e-mail o en foros y listas de discusión.

3.3 Formación técnico-específica

La formación específica del Técnico en Programación es la directamente relacionada con las actividades propias de su Perfil Profesional, por ello los contenidos correspondientes a este campo están agrupados en forma tal que puedan relacionarse fácilmente con las actividades propias del técnico. Para poner en perspectiva y señalar el nivel de los contenidos, se los acompaña con ejemplos de ejercicios prácticos que contribuyan a la formación a través de desempeños que preparen al estudiante para su trabajo futuro.

Las áreas de la formación técnica específica que están relacionadas con la formación del técnico en programación son:

Aspectos formativos referidos a interpretar especificaciones en el contexto de un proyecto Relativos a interpretar críticamente especificaciones.

El software de aplicaciones resuelve necesidades de información o automatización acordadas con usuarios u otros interesados, las que son plasmadas en especificaciones de requerimientos, ya sean estas formales o informales. Programas, subsistemas y otros artefactos de software tienen que diseñarse satisfaciendo esas especificaciones y respetando buenas prácticas, así como manteniendo coherencia con la arquitectura del sistema de software en el que estarán insertos. Estas especificaciones se refieren a las funciones que debe realizar el software, a interacciones con usuarios y otros sistemas, requisitos de calidad y comportamiento y son el punto de partida para lo que va a desarrollar. El programador de software debe ser capaz de interpretarlas, analizándolas críticamente, detectar posibles lagunas o incoherencias y validar su propia interpretación con quienes lideran el proyecto

Contenidos relacionados al análisis y especificación de software:

Requerimientos de software, el proceso, partes interesadas. Requerimientos funcionales, prioridades y criterios de realización. Requerimientos no funcionales, ejemplos y su influencia en el diseño del software. Análisis orientado a objetos y UML. Diagramas de clase. Escenarios, historias y casos de uso; diseño centrado en el usuario. Representación del comportamiento: diagramas de secuencia, máquinas de estado, diagramas de actividad.

Análisis de datos: datos de referencia y de operaciones; de nivel de recursos y de volumen de actividad, diccionario de datos. Organización de datos: modelo Entidad/Relación, principales Formas Normales. Herramientas de modelización. Validación de requerimientos.

Como parte de la forma de adquirir estos aprendizajes y demostración práctica de los resultados alcanzados, en el curso de su formación los estudiantes tienen que:

Interpretar y producir diagramas de clase a partir de problemas sencillos correspondientes a diversos dominios. Analizar y discutir sus propiedades y corrección. Interpretar y especificar casos de uso básicos a partir de descripciones de situaciones realistas. Interpretar artefactos de software (clases, objetos, métodos, algoritmos, tablas).

IF-2018-31697559-GDEBA-DETECDGCYE

Aspectos formativos referidos a planificar su propio trabajo en el contexto de un proyecto, identificando posibles dificultades y organizando sus actividades para encarar la solución del problema planteado

Relativos a la planificación personal dentro de un contexto de proyectos encarados por medio de procesos de ingeniería de software

El programador tiene que identificar aspectos de posible dificultad o riesgo del problema a enfrentar que requieran consulta o un cuidado mayor, evaluando a priori la magnitud del esfuerzo requerido para lograr su solución y considerar la posibilidad de subdividir la asignación en pasos o componentes menores. Esto le permite establecer informalmente un orden o secuencia de trabajo y anticipar la posibilidad de cumplir en tiempo y forma con lo requerido.

Por otra parte, tiene que desenvolverse en el marco de un equipo de trabajo organizado en función del proyecto a encarar. En consecuencia, asume responsabilidad por su asignación dentro del proyecto pero interactúa con pares y líderes del equipo para lograr un mejor proceso conjunto. Ello implica pedir y recibir ayuda para encarar su problema.

En consecuencia, además de conocer y aplicar debidamente las técnicas con que va a realizar la parte asignada, tiene que tener una comprensión del sistema y de la totalidad del proceso, tiene que comprender y cumplir estándares establecidos para el proyecto tratando de aportar lo mejor de su parte, aceptar soluciones resueltas grupalmente o por líderes o especialistas y tiene que colaborar con otros pares y juniors en la solución de los problemas.

Contenidos relacionados con el proceso de ingeniería de software

Conceptos de dinámica de grupos, grupo y equipos de trabajo, características distintivas. La tarea como eje de la convocatoria de todo grupo/equipo. Tarea explícita e implícita.

Dinámica de lo grupal. La mutua representación interna, espacio y tiempo. Objetivos grupales y metas individuales. Lo individual versus lo grupal. Roles y estereotipos, rotación de roles. La comunicación, medios, ruidos que afectan a la comunicación. Importancia de la retroalimentación. La empatía. La escucha activa. Conflictos, técnicas de resolución alternativa.

El equipo de proyectos de desarrollo de software, roles y responsabilidades de sus integrantes. Programas de trabajo y resolución conjunta de problemas. Modelos de ciclo de vida del software y de procesos de desarrollo. El problema del mantenimiento y las migraciones de plataforma.

Metodologías tradicionales y ágiles. Metodologías ágiles, concepto de sprint, fraccionamiento del producto en unidades realizables en un sprint, cola de pendientes, mejora de productos provisionarios (refactoring), variación de los roles y la documentación en el marco de un proceso en el que se aplican metodologías ágiles.

Gestión de los cambios, conceptos de versión, build, producto de la asignación. Concepto de componente. Elementos de administración de la configuración y control de versiones de software. Herramientas de versionado. Otras herramientas (bibliotecas, diccionarios, repositorios) del proyecto.

Conceptos básicos de aseguramiento de la calidad y elementos de métricas. Modelos de madurez de la capacidad de desarrollo. Enfoques para la mejora del proceso. El proceso personal de software, estadísticas personales y capitalización de experiencias.

IF-2018-31697559-GDEBA-DETECDGCYE

Como parte de la forma de adquirir estos aprendizajes y para demostrar prácticamente los resultados alcanzados, en el curso de su formación los estudiantes tienen que realizar:

Participar de proyectos conjuntos de desarrollo de artefactos de software en los que se pongan en práctica diferentes metodologías. Poner en práctica estadísticas elementales propias del proceso personal de software. Realizar revisiones cruzadas de código proponiendo mejoras. Organizar la documentación de un proyecto. Utilizar herramientas de versionado y administración de la configuración. Reflexionar en forma conjunta sobre experiencias y conclusiones obtenidas.

Relativos a la resolución de problemas y al diseño

Los programas, subsistemas y otros artefactos de software tienen que diseñarse respetando buenas prácticas y manteniendo coherencia con la arquitectura existente o prevista del sistema de software en el que estarán insertos o tendrán que interactuar.

Lograr esto requiere no sólo conocer técnicas de diseño de software sino también comprender principios de arquitectura de sistemas de software, propiedades de calidad del software y técnicas de representación.

Contenidos relacionados al diseño de software:

Estrategias de resolución de distintos tipos de problemas. Heurísticas. Principios generales de diseño: descomposición, desacoplamiento, cohesión, reuso, reusabilidad, portabilidad, testeabilidad, flexibilidad, escalabilidad. Diseño estructurado. Diseño orientado a objetos.

Patrones de diseño. Desarrollo de prototipos rápidos para demostración (Rapid prototyping).

Elementos de arquitecturas de software: concepto de vistas, arquitecturas distribuidas, pipe-and-filter, model-view-controller. Diseño orientado al reuso de componentes, incorporación de elementos disponibles al diseño. Diseño de interfases con el usuario.

Como parte de la forma de adquirir estos aprendizajes y demostración práctica de los resultados alcanzados, en el curso de su formación los estudiantes tienen que:

Resolver diversos tipos de problemas. Construir prototipos rápidos con herramientas sencillas. Diseñar soluciones a problemas dados. Diseñar tablas y bases de datos relacionales. A partir de un diseño, analizar clases de equivalencia y diseñar esquemas de prueba. Analizar críticamente la eficiencia y mantenibilidad de diseños alternativos.

Relacionar situaciones dadas con patrones básicos de diseño. Analizar algunos tipos de arquitectura de sistemas de software, discutiendo sus propiedades de calidad (escalabilidad, portabilidad, seguridad, mantenibilidad). Analizar y discutir su eficiencia y escalabilidad.

Aspecto formativo referido a producir y depurar los programas, módulos, clases o subsistemas que respondan a lo requerido, aplicando patrones o reutilizando código en la medida en que resulte posible.

Esto incluye revisar código propio o ajeno para corregir defectos, optimizarlo o adaptarlo a nuevas funcionalidades requeridas. Al hacer esto se aplican buenas prácticas de programación y documentación, conforme a procedimientos de calidad establecidos. También se participa en revisiones cruzadas de artefactos de software para asegurar la calidad del producto.

Relativos a la redacción y depuración de código que responda al diseño propuesto

El programador tiene que construir programas que satisfagan efectiva y eficientemente los requisitos planteados. Estos deben enmarcarse dentro de la arquitectura prevista para el sistema, responder a buenas prácticas, siendo comprensibles y fáciles de modificar, y presentar robustez ante situaciones no previstas. Esto se logra no sólo redactando código, hay que encontrar y adaptar módulos o clases ya existentes para utilizarlas en lo que se está construyendo, verificar lo construido mediante diverso tipo de pruebas y volver a trabajar sobre lo hecho para depurar errores o malfuncionamientos encontrados, así como para optimizarlo.

Por otra parte, nuevos negocios, necesidades de usuarios o regulaciones de las autoridades plantean la necesidad de modificar aplicaciones existentes, con lo cual algún desarrollador tiene que tomar ese programa, interpretar su código para comprenderlo y ubicar dónde ese programa realiza lo que hay que cambiar. Una vez localizado el punto a modificar, tiene que plantear la forma de resolver la situación e introducir los cambios necesarios, probándolo nuevamente para verificar que haga lo esperado y que tampoco hayan cambiado funcionalidades que tenía previamente.

Esto implica programar aplicando conceptos de abstracción, descomposición, algoritmia, estructuras de datos, recursividad, herencia y polimorfismo. Por otra parte, aplicar buenas prácticas de programación y documentación, conocimientos de testing unitario y tener conciencia no sólo del proceso completo de desarrollo, lo que es independiente de la tecnología utilizada, sino también que los programas perduran y van a tener que ser comprendidos y mantenidos por otros.

Contenidos relacionados a algoritmos y estructuras de datos:

Elementos de programación: Estructuras de control. Algoritmos fundamentales, algoritmos numéricos simples. Variables y estructuras de datos estáticas: representación de datos numéricos, rango, precisión y errores de redondeo; arreglos; representación de datos de caracteres, listas y su procesamiento.

Estándares de nomenclatura y formato en programación, encabezado de módulos u objetos con comentarios que expliciten sus alcances y limitaciones, inserción de comentarios o advertencias en el código, documentación adicional.

Estructuras dinámicas. Manejo de memoria en tiempo de ejecución, punteros y referencias, estructuras encadenadas, pilas, colas y hashing. Recolección de espacios no utilizados. La elección de una estructura de datos adecuada.

Diseño orientado a objetos, encapsulamiento y ocultamiento de información, separación entre comportamiento e implementación, clases y subclases, herencia (sustitución), polimorfismo (subtipos vs. herencia), jerarquías de clases, clases colección y protocolos de iteración.

Verificación unitaria de unidades de código, concepto de cubrimiento, organización, ejecución y documentación de la prueba.

Recursión, concepto, funciones matemáticas recursivas, funciones recursivas simples, estrategia de dividir y conquistar, backtracking recursivo. Concepto de autómatas. Elementos de complejidad de algoritmos.

Declaraciones y tipos, la concepción de tipos como conjunto de valores junto con operaciones, modelos de declaración, elementos de verificación de tipos, tipos y polimorfismo paramétrico.

Algoritmos de búsqueda sucesiva y binaria, de ordenamiento con tiempos cuadráticos (selección, inserción), con tiempos $O(N\log N)$ (quicksort, heapsort, mergesort).

Estructuras dinámicas no lineales. Tablas de hashing, estrategias para evitar colisiones.

Árboles de búsqueda binaria, operaciones básicas (búsqueda, inserción y eliminación de nodos). Representación de grafos. Algoritmos de camino mínimo.

Programación conducida por eventos, métodos para manejo de eventos, propagación de eventos, manejo de excepciones.

Integración de imágenes y sonido: estándares más comunes y sus características.

Introducción a la multimedia: concepto, componentes, características, herramientas.

Hipertextos, hipermedios. Animaciones: guías e interpolación de movimientos.

Máquinas virtuales, concepto, jerarquía de máquinas virtuales, lenguajes intermedios, asuntos de seguridad que surgen al ejecutar código en una máquina diferente.

Como parte de la forma de adquirir estos aprendizajes y para demostrar prácticamente los resultados alcanzados, en el curso de su formación los estudiantes tienen que realizar:

Resolver ejercicios de programación, tanto con lápiz y papel como en computador, haciendo hincapié en formalizar el problema y ensayar el enfoque de su solución antes de proceder a la escritura de código, así como en verificar la corrección de la solución obtenida.

Se espera que al concluir el ciclo los estudiantes dominen al menos dos de los tres paradigmas de programación (objetos, imperativa-estructurada o funcional) y varios lenguajes (por lo menos uno correspondiente a cada paradigma, pero también otros, en particular los que tienen aplicación en páginas web). (Se entiende que el tener que adaptarse a diversos tipos de lenguajes de programación y resolver diversa clase de problemas utilizándolos ayuda al proceso de desarrollar capacidad de abstracción.)

Revisar y corregir programas dados. Resolver diversos tipos de problemas comenzando por especificar su propia comprensión de la asignación, diseñar una solución, programar o integrar partes de código utilizando el ambiente de programación indicado, documentándola de acuerdo a buenas prácticas y realizar la verificación unitaria de lo realizado. Intercambiar artefactos de software asumiendo la obligación de interpretar y criticar o mejorar lo recibido.

Desarrollar proyectos grupales durante los cuales se simulen condiciones similares a las del trabajo profesional y en los que cada uno aporte componentes que deben integrarse en el producto final.

Relativos a desarrollar software que utilice bases de datos

El código de los programas se utiliza para computar datos, los que pueden ser internos del programa o, más generalmente, encontrarse o tener que ser almacenados en archivos y bases de datos. En consecuencia, el desarrollador no sólo tiene que conocer de algoritmos y lenguajes, sino también de manejo de la información.

Esto implica conocer de modelos de información que faciliten su almacenamiento y recupero, modelos de datos, indexación, lenguajes de consulta y características de los principales modelos y sistemas de bases de datos.

Actualmente, con sistemas de información distribuidos hace falta obtener o intercambiar datos con otros sistemas a través de Internet y, eventualmente, hacer uso o interactuar con herramientas externas de búsqueda.

Contenidos relacionados con bases de datos:

Modelización de datos, modelos conceptuales (E/R, UML), modelo orientado a objetos, modelo relacional, modelos semiestructurados (XML). Concepto y evolución de los sistemas de bases de datos, sus componentes, funciones de un sistema de base de datos.

Lenguajes de consulta (SQL, QBE), definición de datos, álgebra relacional, formulación de consultas, lenguaje de actualización, restricciones, integridad. SQL embebido en un lenguaje imperativo. Scripts. Introducción a un lenguaje de consulta de objetos.

Procedimientos almacenados.

Diseño de bases de datos, dependencia funcional, formas normales, descomposición de un esquema, claves primarias y secundarias. Procesamiento de transacciones, fallas y recuperación, control de concurrencia. Bases de datos distribuidas, problemas que surgen con su explotación. Problemas de escalabilidad, eficiencia y efectividad. Privacidad, integridad, seguridad y preservación de la información.

Como parte de la forma de adquirir estos aprendizajes y para demostrar prácticamente los resultados alcanzados, en el curso de su formación los estudiantes tienen que realizar:

Resolver ejercicios de álgebra relacional. Se espera que al concluir el ciclo los estudiantes resulten capaces de explotar una base de datos relacional. Revisar y corregir programas dados. Resolver diversos tipos de problemas comenzando por especificar consultas a bases de datos dadas, programar actualizaciones de datos en base a cálculos con nuevos datos, preocupándose tanto por la integridad de la información como por la eficiencia de los procesos. Diseñar tablas y bases de datos, incorporar procedimientos. Desarrollar proyectos grupales durante los cuales se simulen condiciones similares a las del trabajo profesional y cada uno aporte componentes que deben integrarse en el producto final.

Relativos a producir interfaces adecuadas para el usuario

En los sistemas de información el usuario suele proveer datos al sistema y utilizar la información que brinda el sistema para tomar decisiones de diverso tipo. En tal sentido, la calidad de las interfaces y la interacción del usuario con el sistema resultan muy importantes, ya que interfases pobremente diseñadas pueden llevar a registrar mal los datos o a dificultar el uso del sistema. En particular, cuando se producen situaciones de excepción (datos o comandos incorrectos por parte del usuario o la solicitud de algo que el sistema no puede realizar) es conveniente planificar un diálogo adecuado para resolver la situación, incluyendo ayudas para el usuario.

En consecuencia, el desarrollador, a pesar de que inscriba su componente en un diseño más general, debe conocer distintos tipos de interfases con el usuario, principios de diseño de interfases visuales, verificaciones básicas a realizar sobre los datos de entrada y manejo de ayudas y del diálogo para superar las dificultades que pueda encontrar el usuario.

En la actualidad se han difundido una serie de dispositivos (móviles, GPSs, tabletas de diversas características, pantallas que reaccionan al contacto, recolectores de datos) que amplían el espectro de las interfases con los usuarios, lo que genera una gama de tecnologías y modelos de interacción que un buen desarrollador de software debe conocer para su trabajo o, al menos, estar en condiciones de adaptarse rápidamente.

IF-2018-31697559-GDEBA-DETECDGCYE

Contenidos relacionados con interacción ser humano-máquina:

Interacción ser humano-máquina, conceptos básicos. Distintos contextos para interfaces: visuales o de texto en aplicaciones habituales, interfaces web con dispositivos para navegación, sistemas colaborativos, juegos y otras aplicaciones multimediales, interfaces con o por medio de diversos dispositivos, lo que pueden incluir teléfonos móviles y TV digital.

Proceso de desarrollo centrado en el usuario: foco temprano en los usuarios, prueba empírica de la calidad, diseño iterativo. Medidas de evaluación: utilidad, eficiencia, facilidad de aprendizaje, satisfacción del usuario. Modelos de diseño de la interacción: atención, movimiento, cognición, percepción y reconocimiento.

Diseño para el cambio: adaptación a otras lenguas o localismos, adaptación a la diversidad de condiciones humanas. Notación para especificar interfaces. El manejo de los errores del usuario o del sistema. Técnicas y herramientas de prototipado.

Principios de interfaces gráficas, acción-objeto vs. objeto-acción, eventos en interfaces de usuario, estándares, errores más comunes. Interfaces para un sistema nativo, uso del browser para sistemas que operen en la web.

Como parte de la forma de adquirir estos aprendizajes y para demostrar prácticamente los resultados alcanzados, en el curso de su formación los estudiantes tienen que realizar:

Considerar, discutir y diseñar interacciones software-usuario. Diseñar diversas pantallas que respondan a determinadas propuestas y evaluar conjuntamente lo obtenido. Diseñar interfaces para la web con ayudas para la navegación. Diseñar interfaces para alguna norma estándar (USB, bluetooth) para dispositivos.

Relativos a desarrollar software que opere en ambientes distribuidos

En la actualidad la mayor parte de los sistemas operan en forma distribuida a través de redes locales o de Internet, utilizando en muchos casos como interfase el software de navegación (browser) de la máquina cliente. Esto implica mantener un diálogo cliente-servidor que intercambie datos y permita acceder y actualizar bases de datos situadas a distancia.

El desarrollador tiene que conocer y poner en práctica la programación en un ambiente clienteservidor, para lo cual tiene que comprender conceptos de arquitectura de sistemas web, aspectos de seguridad y de comportamiento.

Contenidos relacionados con computación centrada en redes:

Aplicaciones en redes. Protocolos a nivel de la capa de aplicación. Interfaces web: browsers y APIs. Subprotocolos TCP y UDP. El socket como abstracción.

Modelo cliente servidor. Primeras acciones de ambos. Creación de sockets y ligado de direcciones. Par cliente/servidor TCP. Concepto de sesión. Par cliente/servidor UDP.

Concepto de paquete. Polling con primitivas bloqueantes. RPC. Protocolos web. Lenguajes de programación utilizados para el desarrollo de páginas y sistemas web.

Principios de ingeniería web. Sitios web estructurados mediante bases de datos.

Tecnologías de búsqueda en web. El papel del middleware, herramientas de apoyo.

Aplicaciones cooperativas. Sistemas de workflow. Herramientas para desarrollo en ambientes web. Frameworks de aplicaciones y su utilización.

Creación y administración de sitios web.

Como parte de la forma de adquirir estos aprendizajes y para demostrar prácticamente los resultados alcanzados, en el curso de su formación los estudiantes tienen que realizar:

Diseñar páginas web estáticas y dinámicas. Diseñar y programar aplicaciones sencillas que interactúen en un ambiente cliente-servidor. Diseñar sitios web organizados como bases de datos para que el usuario pueda actualizarlos sin intervención de desarrolladores. Utilizar ambientes de programación para web, programar aplicaciones interactivas que actualicen bases de datos, considerar y discutir aspectos de seguridad relativos a las mismas.

Verificar los artefactos de software contruidos considerando las necesidades de cobertura de la prueba.

Para ello diseña los casos considerando el entorno de pruebas y ejecuta pruebas unitarias, así como registra los datos y resultados. De ser necesario, realiza acciones correctivas sobre el código hasta satisfacerse de que cumpla con las especificaciones recibidas.

Relativos a verificar el buen funcionamiento de los programas

En el software es tan alta la distancia entre el diseño y la construcción, que resulta totalmente improbable producir inicialmente programas sin defectos. Así es que los productos tienen que ser verificados mediante pruebas que comprueben su calidad. Para ello hay que diseñar conjuntos de datos de prueba y realizar procesos en condiciones controladas que den cuenta de diversos aspectos. En primer lugar, que el programa satisfaga los requerimientos planteados. También que tenga robustez y no acepte datos incorrectos o no realice acciones imprevistas cuando un usuario se equivoca en un comando.

Esto se inscribe en el concepto de verificación unitaria, que debe realizar el desarrollador de software para satisfacerse que ha realizado lo requerido. Sin embargo, la buena práctica implica que un grupo independiente debe integrar lo realizado por cada desarrollador y someterlo a prueba conjunta, lo que puede poner de relieve fallas originadas en la interacción.

La detección de fallas motiva que el desarrollador vuelva sobre el código para encontrar los defectos y los resuelva.

Para aplicar con propiedad técnicas de testing un desarrollador de software tiene que conocer principios generales, los diversos tipos de testing que se utilizan en el proceso de desarrollo de software y ser capaz de utilizar apropiadamente ambientes y herramientas específicos de testing unitario.

Contenidos relacionados con testing

Distinción entre validación y verificación. Enfoques estáticos y dinámicos. Fundamentos de testing, testeo de caja negra y de caja blanca. Pruebas funcionales: generación de casos o datos de prueba, clases de equivalencia. Pruebas estructurales: pruebas estáticas, pruebas dinámicas, cobertura de la prueba. Otro tipo de objetivos: verificación de usabilidad, confiabilidad, seguridad. Registro de fallas e informes técnicos.

Prueba unitaria, de integración, validación y prueba del sistema. Desarrollo conducido por el testeo. Refactorización del código. Testeo de regresión. Verificación y validación de artefactos

que no constituyen código: documentación, archivos de ayuda, material de capacitación. Inspecciones, revisiones cruzadas, auditorías.

Como parte de la forma de adquirir estos aprendizajes y para demostrar prácticamente los resultados alcanzados, en el curso de su formación los estudiantes tienen que realizar:

- Procesar pruebas e identificar defectos en artefactos producidos por sí mismos o por otros. Planificar y diseñar casos y conjuntos de datos para prueba de artefactos dados, respondiendo a objetivos y requisitos de cobertura. Implementar pruebas unitarias de programas y pequeños sistemas utilizando herramientas y creando los ambientes necesarios, realizar los procesos y revisar los resultados para generar informes de fallas.

Desempeño de base

Conocer y saber utilizar con propiedad y en condiciones de seguridad recursos de hardware, software y redes para emplear los ambientes que necesite para el desarrollo y la verificación del software, mantener los repositorios de información que necesite utilizar y disponer de los productos de su trabajo en condiciones de confiabilidad.

Relativos al ambiente de desarrollo

El desarrollador no sólo tiene que tener capacidades como para resolver los problemas que presenta el diseñar y programar artefactos de software que satisfagan las asignaciones planteadas en el contexto de la arquitectura propuesta. Tiene que configurar el ambiente de programación y el de testing que va a utilizar en su trabajo, generar o extraer datos para producir los que necesite para probar lo que realizó. Eventualmente, tratar de interpretar fallas en función de posibles problemas de compatibilidad con otro software.

Para realizar esto el desarrollador debe conocer sobre sistemas operativos y debe ser capaz de manejarse hábilmente con diversos editores, configurar aspectos de software y hardware y explotar con habilidad recursos de programación, incluyendo entre los mismos bibliotecas de objetos y programas propias, de su organización o disponibles a través de Internet, así como plantear y resolver consultas de problemas de programación a través de foros y listas públicas o privadas.

Contenidos relacionados con sistemas operativos, editores y bibliotecas de programas

- Los sistemas operativos, su papel y propósito, la historia de su desarrollo, funcionalidades típicas. Mecanismos que soportan los modelos cliente-servidor y otros dispositivos.
- Características y objetivos de su diseño y su influencia en la seguridad, interoperabilidad, capacidad multimedial.
- Aplicaciones complementarias (navegadores, defragmentadores, antivirus, traductores de medios audiovisuales).
- Estructuras de sistemas operativos (monolíticos, modulares y de micro kernel).
- Abstracciones, procesos y recursos. Organización de los dispositivos, interrupciones: métodos e implementación. Concepto de estados usuario/supervisor y protección, transición al modo supervisor.
- Estados y transiciones; cola de procesos, bloque de control de procesos. Despacho, switching de contexto, switching cooperativo y preempted. Ejecución concurrente;

IP-2018-31697359-GDEBA-DETECDGCYE

ventajas y desventajas. El problema de la exclusión mutua y algunas soluciones. Bloqueos: causas, condiciones, prevención. Paso de mensajes sincrónico y asincrónico. Problema de consumidor-productor y sincronización (mutex, semáforos). Multiprocesamiento (interrupción de ciclos, reentrada).

- Políticas de despacho de procesos; programación con y sin prioridades de interrupción. Procesos y threads. Elementos de tiempo real y tiempos límite.
- Administración de memoria. Revisión de memoria física y hardware de administración de memoria. Paginamiento y memoria virtual. Working sets y trashing. Cacheo.
- Administración de dispositivos, características de dispositivos seriales y paralelos.
- Abstracción de diferencias entre dispositivos. Estrategias de buffering. Acceso directo a memoria. Recuperación de fallas.
- Representación de caracteres, audio e imágenes. Compresión de datos, códigos para detectar o corregir errores.
- Seguridad y protección. Políticas y mecanismos de separación. Métodos y dispositivos de seguridad. Protección, control de acceso y autenticación. Backups.
- Sistemas de archivo (datos, metadatos, operaciones, organización, buffering, secuenciales y no secuenciales). Índices: contenido y estructura. Técnicas estándares de implementación. Archivos de mapeo de memoria. Sistemas de archivo para propósitos especiales. Denominación, búsqueda, acceso, backups.
- Scripting. Comandos básicos del sistema, creación de scripts, pasaje de parámetros.
- Ejecución de un script.
- Ambientes gráficos para edición, editores inteligentes. Herramientas integradas disponibles para la edición en distintos lenguajes y ambientes. Bibliotecas de clases, programas y rutinas.
- Aspectos de administración de redes, uso de contraseñas y mecanismos de control de acceso, servidores de nombres de dominios y de servicios, proveedores de servicios en Internet. Aspectos de seguridad y firewalls. Asuntos de calidad de servicio: comportamiento, recuperación de fallos.

Como parte de la forma de adquirir estos aprendizajes y demostración práctica de los resultados alcanzados, en el curso de su formación los estudiantes tienen que realizar:

- Localizar y seleccionar artefactos de software, libre o bajo otras licencias, que respondan a ciertas características. Instalar, configurar y personalizar sistemas operativos, compiladores de lenguajes, editores y otros ambientes de programación o de prueba de programas. Crear y organizar repositorios de documentación y programas para uso personal o de proyectos.

<div> <div>4° AÑO</div> <div>FORMACIÓN CIENTÍFICO TECNOLÓGICA</div> </div> <div> <div>MATEMÁTICA CICLO SUPERIOR</div> </div>	
<div>CARGA HORARIA</div>	<div> <div>4 Horas Semanales</div> <div>144 Horas Anuales</div> </div>
<div> <div>CAPACIDADES A DESARROLLAR</div> </div>	<div> <p>Finalizado el cursado de la materia los y las estudiantes deberán haber desarrollado las siguientes Capacidades:</p> <ul style="list-style-type: none"> - Construir conocimientos matemáticos significativos. - Reconozca y aplique las propiedades de las funciones trigonométricas - Analizar funciones a partir de sus gráficas. - Interpretar soluciones gráficas y analíticas. - Establecer transferencias pertinentes de los conceptos a situaciones intra y/o extra-matemáticas de la especialidad. - Trabajar de manera autónoma identificando posibles modelizaciones de situaciones que se presenten en diferentes campos. - Comprobar la importancia de la formalización como herramienta de comunicación en el ámbito de la Matemática. - Utilizar software de aplicación en modelos matemáticos. - Aplicar de manera autónoma diversas estrategias en la resolución de problemas. </div>
	<div> <div>CONTENIDOS</div> </div>
	<div> <p>Números Reales y Números Complejos: Concepto y representación. Completitud. Operatoria. Sucesiones aritméticas, geométricas y otras. Notación y lenguaje. Funciones: polinómicas, valor absoluto, exponencial, logarítmicas, racionales, trigonométricas. Operaciones con funciones. Función inversa. Reconocimiento y caracterización de funciones desde su gráfica y su fórmula: intervalos de crecimiento y decrecimiento, máximos y</p> </div>

mínimos, periodicidad, continuidad, paridad, ceros. Semejanza de figuras planas Teorema de Thales. Resolución de triángulos mediante el empleo de la trigonometría: teoremas del seno y del coseno. Modelización de fenómenos del mundo real a través del empleo de funciones. Ecuaciones e inecuaciones. Sistemas de ecuaciones. Formas de resolución gráfica y analítica. Resolución de situaciones problemáticas modelizadas. Comparación de métodos de resolución y discusión del número y tipo de soluciones halladas de acuerdo a los contextos de las situaciones a resolver

ORIENTACIONES DIDÁCTICAS

- Resolución de situaciones problemáticas vinculadas a los contenidos planteados.
- Realización e interpretación de gráficos y tablas.
- Utilización de algoritmos para efectuar operaciones.
- Buceo bibliográfico. Criterios de evaluación:
- Evaluación integrada y en proceso.
- Comprensión e interpretación de conceptos específicos del espacio curricular.
- Comprensión y explicación de los problemas planteados para interpretar la realidad matemática que nos rodea.
- Síntesis y análisis de resultados.

Criterios de Evaluación

- Evaluación integrada y en proceso.
- Comprensión e interpretación de conceptos específicos del espacio curricular.
- Comprensión y explicación de los problemas planteados para interpretar la realidad matemática que nos rodea.
- Síntesis y análisis de resultados.
- Se pretende que el estudiante evidencie un cambio actitudinal y procedimental en función del recorrido en el ciclo lectivo

<div> <div>4° AÑO</div> <div>FORMACIÓN CIENTÍFICO TECNOLÓGICA</div> </div> <div>FÍSICA</div>	
CARGA HORARIA	<div> <div>3 Horas Semanales</div> <div>108 Horas Anuales</div> </div>
CAPACIDADES A DESARROLLAR	<p>Finalizado el cursado de la materia los y las estudiantes deberán haber desarrollado las siguientes Capacidades:</p> <ul style="list-style-type: none"> - Incorporar de forma empírica los conceptos físicos de la naturaleza. - Ubicar un punto en el plano y hallar su posición, también aplicar escalas y unidades métricas. - Interpretar problemas y leer gráficos de movimientos y resolverlos. - Utilizar distintos recursos para la investigación. - Expresar en forma oral y grupal las distintas leyes de Newton. - Visualizar las propiedades físicas de la materia y de los cuerpos. - Lograr utilizar elementos de laboratorio.
	CONTENIDOS
	<p>Energía mecánica: cinemática, dinámica. Transformaciones de energía. Energía térmica: calor y temperatura. Transferencia de energía. Escalas termométricas. Calorimetría. Dilatación de líquidos y sólidos. Energía Eléctrica: campo y potencial electrostáticos. Campo Magnético. Corrientes eléctricas variables. Circuitos eléctricos. Óptica geométrica. Propagación de la luz, ondas y rayos, reflexión y refracción, espejos y lentes cóncavas y convexas, marcha de rayos. Lentes y sus aberraciones. Acústica. Mecanismo de propagación y distribución del sonido, ondas sonoras e intensidad, aislamiento, absorción, reflexión, reverberación. Efecto Doppler. Supresión de ruido.</p>

ORIENTACIONES DIDÁCTICAS

- Experimentación de distintos fenómenos físicos
- Reducción y uso de unidades, pasajes de unidades. Equivalencia. Ubicación en el plano y construcción de una brújula.
- Confección y uso de hoja de fórmulas como herramienta en la resolución de problemas.
- Uso de gráficos de movimientos y utilización de software de simulación.
- Aplicación de elementos geométricos.
- Utilización de materiales de laboratorio.
- Utilización de medios audiovisuales.
- Aplicación de metodología de investigación y realización de proyecto científicos.

CARGA HORARIA
2 Horas Semanales
72 Horas Anuales

Finalizado el cursado de la materia los y las estudiantes deberán haber desarrollado las siguientes Capacidades:

- Presentar la información utilizando símbolos y anotaciones técnica
- Realizar cálculos estequiométricos utilizando pesaje de reactivos
- Diseñar pasajes necesarios para preparar soluciones y llevar a la practica en el laboratorio
- Describir y analizar comparativamente los distintos procesos
- Calcular cantidades para la formulación de reacciones y la preparación de soluciones
- Explicitar correctamente los resultados.
- Resolver problemas de soluciones de baja complejidad.
- Utilizar normas de procedimiento, calidad, seguridad y protección ambiental.
- Manejar la tabla periódica de los elementos.
- Identificar las propiedades de los distintos grupos de la tabla periódica.
- Interpretar fórmulas y nomenclatura de diferentes sustancias.
- Representar reacciones mediante ecuaciones químicas

**CAPACIDADES A
DESARROLLAR**

CONTENIDOS	
Estructura atómica. Tabla periódica. Uniones Químicas. Funciones inorgánica. Ecuaciones químicas. Estequiometría. Gases: leyes. Soluciones. Química en procesos industriales: reacciones de óxido – reducción. Reacciones de ácido – base. pH. Procesos de equilibrio. Electroquímica. Pilas. Funciones orgánica. Química y combustibles. Destilación. Refinación. Fuentes de energía y combustibles alternativos. Normas de seguridad para el trabajo en el laboratorio.	
ORIENTACIONES DIDÁCTICAS	
<ul style="list-style-type: none"> - Hablar leer y escribir en químicas, la comunicación directa es una actividad central para el desarrollo científico - Identificar problemas y sus conexiones conceptuales - Trabajar los contenidos de la materia a partir de situaciones problemáticas - Trabajar con modelos 	

4° AÑO
FORMACIÓN
CIENTÍFICO
TECNOLÓGICA

ARQUITECTURA DE HARDWARE

CARGA HORARIA

2 Horas Semanales
72 Horas Anuales

Finalizado el cursado de la materia los y las estudiantes deberán haber desarrollado las siguientes Capacidades:

- Valorar el uso de los materiales y simbología electrónica.
- Identificar unidades de medida.
- Caracterizar los elementos que intervienen en los circuitos.
- Utilizar protoboard para la confección de diversos circuitos y el uso de componentes habituales en la electrónica.
- Adquirir habilidades sobre el uso de múltiples plataformas

CAPACIDADES A
DESARROLLAR

CONTENIDOS

Fundamentos de la Electricidad. Corriente eléctrica: concepto. Diferencia de potencial. Conductividad. Resistencia. Ley de Ohm. Potencia eléctrica. Circuitos serie y paralelo. Corriente Continua. Circuitos de corriente continua. Circuitos de una malla. Leyes de Kirchoff. Circuitos de varias mallas. Corriente alterna: concepto. Tensiones variables. Trenes de pulso. Valores: máximo, medio y eficaz. Período.

Frecuencia. Tiempo de subida y bajada. Electrostática. Dipolo eléctrico. Estructura molecular de los dieléctricos. Polarización del dieléctrico. Inducción electrostática en un conductor. Capacitores. Tipos y usos. Efecto del dieléctrico sobre el valor de capacidad. Asociación de capacitores. Circuitos magnéticos. Transformadores. Áreas de aplicación. Semiconductores. Materiales semiconductores. Conducción por huecos y electrones. Material intrínseco. Dopaje. Materiales N y P. Unión PN. Dispositivos Electrónicos. Dispositivos electrónicos de estado sólido. Unión PN como

rectificador. Diodos. Fuentes lineales con filtro por capacitor. Diodo de ruptura. Transistor de juntura. Transistores de efecto de campo, MOS y MOSFET de potencia. Usos del transistor como amplificador y como conmutador, Fuentes switching, otros. Lectura e interpretación de DataSheet Usos del protoboard para la confección de diversos circuitos y el uso de componentes habituales en la electrónica. La incorporación de tecnologías como arduino raspberry, o tecnologías similares adaptadas al mercado local

ORIENTACIONES DIDÁCTICAS	
	<ul style="list-style-type: none"> - Resolución de ejercicios y problemas - Adquisición de aprendizajes mediante el estudio de casos reales o simulados - Realización de un proyecto para la resolución de un problema aplicando habilidades y conocimientos adquiridos - Debate, diálogo, exposición, demostración. - Uso de simuladores. - Uso de tecnologías disponibles en el establecimiento

<div>4° AÑO</div> <div>FORMACIÓN TÉCNICA ESPECÍFICA</div> <div>METODOLOGIAS DE PROGRAMACIÓN</div>	
CARGA HORARIA	<div>2 Horas Semanales</div> <div>72 Horas Anuales</div>
CAPACIDADES A DESARROLLAR	<p>Finalizado el cursado de la materia los y las estudiantes deberán haber desarrollado las siguientes Capacidades:</p> <ul style="list-style-type: none"> - Interpretar y comprender problemas factibles de ser sistematizados. - Abstraer modelos conceptuales. - Desarrollar esquemas de razonamiento lógico en la resolución de problemas. - Identificar y clasificar los datos de la futura solución. - Desarrollar algoritmos elementales. - Diseñar, desarrollar e implementar soluciones informáticas simples.
CONTENIDOS	
<p>Recomienda el trabajo sobre el ANSI C/C++. Se recomienda la documentación en UML. Como casos de Uso, diagrama de dominio. Se aborda Confección de la documentación. Al comentar Ventajas de comentar los programas realizados. Implementación y seguimiento de la solución desarrollada. Asistencia básica al usuario.</p> <p>Que es un programa y herramientas de programador:</p> <p>Entornos de desarrollo e IDE'S desarrollo.</p> <p>Concepto de código fuente, objeto y binario.</p> <p>Estructuras: Estructuras de repetición. For ..., do while..., if ..., foreach..., select case, etc.</p>	

Estructuras condicionales. Simples, anidadas, condicionales lógicas.

Algoritmos de resolución mediante métodos lineales. Aplicación del criterio top down en la resolución de pequeños problemas.

Concepto de variable y constante. Identificación del tipo de dato. Concepto de Contador y Acumulador. Tipos de datos y modificadores. Tipos de datos aplicados a la programación. Determinación del tipo de dato. Variables enteras, reales booleanas, doble precisión y de caracteres. Rango de datos. Variables locales y globales. Modificadores de tipos. Palabras reservadas.

Operadores y su precedencia. Operadores aritméticos, relacionales y lógicos. Operadores unarios. Operadores a nivel de bytes y a nivel de bits. Precedencia de operadores.

Condiciones de corte y salida de programa. Diseño de programas, técnicas para la construcción, documentación y seguimiento.

Conceptos básicos de librerías. Funciones, procedimientos, devoluciones, arrays, matrices.

Abordaje a través de la incursión a foros, de las temáticas afines al espacio y las situaciones problemáticas.

ORIENTACIONES DIDÁCTICAS

- Las orientaciones se darán en base al contenido a través del:
 - Aprendizaje colaborativo.
 - Uso y análisis de la información.
 - Aprendizaje basado en un proyecto técnico.
 - Exposición en equipo.
 - Presentación en plenaria con el uso de las herramientas.
 - Autoaprendizaje. Investigación
 - Búsqueda y análisis de información

<div>4° AÑO</div> <div>FORMACIÓN TÉCNICA ESPECÍFICA</div> <div>HARDWARE Y COMPONENTES</div>	
CARGA HORARIA	<div>4 Horas Semanales</div> <div>144 Horas Anuales</div>
<div>CAPACIDADES A DESARROLLAR</div>	<div>Finalizado el cursado de la materia los y las estudiantes deberán haber desarrollado las siguientes Capacidades:</div> <ul style="list-style-type: none"> - Soluciones a problemáticas en el ámbito de la tecnología de cómputos. - Buenas prácticas de seguridad e higiene de las personas y equipos
	CONTENIDOS
	<p>Modelo Von Neumann: La Organización y Arquitectura de un sistema computacional. Cómo funciona un sistema computacional. Introducción: Organización básica de un CPU – Registros y Buses – Clocks – Subsistema de Entrada / Salida – Organización y direccionamiento de la memoria. Interrupciones. Ejemplos de arquitecturas reales: Modelo de von Neumann – Modelo de Harvard. Concepto de microprocesador. Refurbishing de equipos. Conectores externos. Conectores internos. Alimentación del motherboard. Componentes integrados. Integración sin cables. Detección de errores, a través de placas de diagnóstico. Falta de tensión. Unidades de almacenamiento. Almacenamiento externo.</p> <p>Motherboard. Elementos del motherboard. Componentes del motherboard. El chipset. El southbridge. Puentes de conexión . Bancos de memoria. Zócalo del procesador. Front Side Bus. Los zócalos modernos. El BIOS. Códigos sonoros del POST. Configurar el BIOS. Actualizar el BIOS. Limpiar el CMOS. Microprocesador. Evolución de los procesadores y velocidades. Las partes de la CPU. Arquitecturas actuales. El cooler, mantenimiento. Los componentes del equipo de refrigeración. Memoria RAM. Cómo funciona. Tipos de memoria. Formatos y características. . Instalación de memoria</p>

RAM. La fuente de alimentación. Las primeras conexiones. Potencia máxima de a fuente. Los conectores de la fuente. Montar la placa madre. Elementos de sujeción. Elementos de Seguridad Personal y de los Equipos. Análisis de tensión y Circuitos. Seguridad en la Instalación Eléctrica. Estabilizadores. Ups. Montaje, configuración.

ORIENTACIONES DIDÁCTICAS

- Revisión de Equipos, reparación.
- Configuración de dispositivos
- Manejo de herramienta de laboratorio.

<div> <div>4° AÑO</div> <div>FORMACIÓN TÉCNICA ESPECÍFICA</div> </div> <div> <div>SISTEMAS OPERATIVOS</div> </div>	
<div>CARGA HORARIA</div>	<div> <div>4 Horas Semanales</div> <div>144 Horas Anuales</div> </div>
<div> <div>CAPACIDADES A DESARROLLAR</div> </div>	<div> <p>Finalizado el cursado de la materia los y las estudiantes deberán haber desarrollado las siguientes Capacidades:</p> <ul style="list-style-type: none"> - Reconocer las funciones de un sistema operativo - Conocer los diferentes comandos para manejo de archivos - Identificar las características de cada sistema operativo. - Configurar equipos monousuarios a través del panel de control. - Personalización de la interfaz de usuario. - Administración de cuentas de usuario </div>
<div>CONTENIDOS</div>	
<div> <p>Esta descripción presenta aquellos contenidos que podrían desarrollarse en el transcurso de las actividades formativas. Será el equipo docente a cargo del módulo quien presente sistemas operativos en dispositivos móviles y de escritorio resuelva en qué momento y a través de qué actividades los desarrollará.</p> <p>Concepto de sistema operativo (sus funciones y estructura, descripción elemental de cómo trabaja y de las herramientas de diagnóstico que brinda). Instalación y manejo del sistema operativo (visualizaciones que brinda, comandos que permiten operar, instalación o actualización de elementos optativos, personalizaciones que ayudan al usuario). Virtualización de sistemas operativos. Introducción a aspectos de conectividad.</p> <p>Concepto de sistema operativo. Manejo de archivos; de entrada/salida; de la memoria. Conceptos de cómo funciona: núcleo, interrupciones y</p> </div>	

llamadas al sistema, su utilidad para situaciones de error. Panel de control, sus posibilidades de configuración. Arranque y carga del sistema operativo.

La función de imprimir en forma diferida al procesamiento. Utilidad práctica de administrar la impresión, programa que lo realiza y comandos que la gobiernan. La utilización de varias impresoras y la preparación de salidas para impresoras no necesariamente conectadas. Controladores de impresión (*drivers*), su instalación y actualización. Tipografías (*fonts*), su instalación y funcionamiento.

La interface con el usuario. Principios de claridad y mejor utilización, posibilidades de organización y apagado de la pantalla. Incidencia en la operatoria y los errores del usuario; cómo personalizarla.

Tipos y organización de la memoria real; su asignación, administración y protección. Formas de administrar la memoria y su incidencia en el comportamiento de la ejecución de programas. Programas residentes. Herramientas para diagnóstico del uso y reasignación de la memoria. Memoria virtual y su administración; el problema del *swapping* y sus efectos sobre la rapidez de ejecución.

Concepto de proceso. Prioridades y esquemas de ejecución, recursos que utilizan los programas. Tipos de interrupción. Formas de recabar información y alterar las prioridades de ejecución, efectos de hacerlo. Procesos con líneas de ejecución paralelas para aprovechar la disponibilidad de múltiples procesadores. Problemas de comportamiento y herramientas para diagnóstico.

La extensión de los límites de la máquina a un grupo o al mundo. Concepto de grupo, acceso a recursos compartidos. Concepto de Internet, browser y funciones que brinda; su interacción con el sistema operativo, problemas de ajuste (*setting*) y compatibilidad con aplicaciones.

Conceptos de seguridad. Instalación del sistema operativo. Parámetros de instalación y posibilidades de configuración en función de la máquina y el entorno de aplicaciones.

Distintos tipos de Arquitecturas. Sistemas abiertos y cerrados o propietarios, características, instalación, comparación, dll's, scripts, sistemas de archivos, compatibilidad, núcleo de un SO, modularidad, etc.

Instalación de sistemas operativos. Análisis de las características de los principales sistemas operativos del mercado.

ORIENTACIONES DIDÁCTICAS

- Debate, diálogo, exposición, demostración.
- Construir conocimientos a través de la interacción y la actividad de los estudiantes con actividades como seminarios, cursos.
- Preparación de ensayos (monografías)
- Lecturas complementarias

<div>4° AÑO</div> <div>FORMACIÓN TÉCNICA ESPECÍFICA</div> <div>SUITE DE APLICACIONES</div>	
CARGA HORARIA	<div>2 Horas Semanales</div> <div>72 Horas Anuales</div>
<div>CAPACIDADES A DESARROLLAR</div>	<p>Finalizado el cursado de la materia los y las estudiantes deberán haber desarrollado las siguientes Capacidades:</p> <ul style="list-style-type: none"> - Utilizar de manera práctica los programas de Procesador de textos, Planilla de Cálculo, Base de datos y de Presentaciones. - Reconocer e identificar las actualizaciones que se producen de acuerdo al avance tecnológico. - Analizar, organizar y resolver cuestiones numéricas, datos financieros, estadísticos y similares. - Detectar situaciones problemáticas y seleccionar la herramienta adecuada para resolverla. - Aprendizaje colaborativo en línea.
	CONTENIDOS
	<p>OFIMÁTICA: Manejo de Hipervínculos, encadenamiento de imágenes. Documentos con hipervínculos. Fórmulas y funciones, macros, métodos abreviados. Fórmulas que vinculen varias hojas. Representación de datos mediante gráficos. Aplicaciones para el diseño y desarrollo de presentaciones. Creación de una presentación dinámica</p> <p>Aplicaciones para la gestión y administración de bases de datos. Concepto de base de datos. Archivo, registros y campos. Motores de base de datos. Creación de una base de datos. Tablas, formularios, consultas e informes. Utilidades de las bases de datos. Seguridad. Diseño y desarrollo de un ABM simple.</p> <p>Conceptos básicos de cloud: Herramientas, accesos, ofertas de software, seguridad y protección de datos, infraestructura, servidores, manejo de</p>

datos, almacenamiento.
ORIENTACIONES DIDÁCTICAS
<ul style="list-style-type: none"> - Aprendizaje colaborativo. - Uso y análisis de la información. - Aprendizaje basado en un proyecto técnico. - Exposición en equipo. - Presentación en plenaria con el uso de las herramientas. - Autoaprendizaje. Investigación - Búsqueda y análisis de información

<p>5° AÑO</p> <p>FORMACIÓN CIENTÍFICO TECNOLÓGICA</p>	<p>ANÁLISIS MATEMÁTICO</p>
<p>CARGA HORARIA</p> <p>4 Horas Semanales 144 Horas Anuales</p>	<p>Finalizado el cursado de la materia los y las estudiantes deberán haber desarrollado las siguientes Capacidades:</p> <ul style="list-style-type: none"> - Interpretar el concepto de límite. - Identificar funciones a partir de su derivada. - Establecer transferencias pertinentes de los conceptos a situaciones intra y/o extra-matemáticas de la especialidad. - Comprobar la importancia de la formalización como herramienta de comunicación en el ámbito de la Matemática. - Aplicar el concepto de integral en el cálculo de áreas y volúmenes. - Comparar las soluciones analíticas y gráficas. - Utilizar software de aplicación en modelos matemáticos. - Aplicar de manera autónoma diversas estrategias en la resolución de problemas. <p>CAPACIDADES A DESARROLLAR</p> <p>CONTENIDOS</p> <p>Sucesiones. Límite. Propiedades. Convergencia y divergencia. Límites por derecha e izquierda. Límite de una función en un punto, en el infinito, por derecha y por izquierda. Continuidad. Derivada de una función en un punto. Derivada de funciones elementales. Ceros de una función. Crecimiento y decrecimiento. Máximos y mínimos. Puntos de inflexión. Gráfica de una función y su derivada. Modelización de fenómenos del mundo real y de otras áreas usando funciones. Integrales indefinidas, racionales, trigonométricas, definidas. Teorema fundamental del cálculo. Integración numérica: regla</p>

de los trapecios y fórmula de Simpson. Aplicaciones. Integral doble, integral triple: definición y propiedades. Aplicaciones Cambio de variables: su aplicación para simplificar el cálculo de integrales. Integrales impropias. Formas indeterminadas. Regla de L'Hôpital

ORIENTACIONES DIDÁCTICAS

- Resolución de situaciones problemáticas vinculadas a la especialidad.
- Organización de datos en tablas que permita la generalización de los resultados
- Utilización de algoritmos para efectuar operaciones.
- La evaluación será integral y en proceso.
- Se evaluará: la comprensión, los procedimientos, la interpretación y la presentación.

Criterios de evaluación

Se pretende que el estudiante evidencie un cambio actitudinal y procedimental en función del recorrido en el ciclo lectivo.

Se verificará a través de la progresión lógica de contenidos y complejidad de estos, siguiendo un esquema propio de las pedagogías constructivas.

El aporte colaborativo será fundamental, sobre todo en los espacios de formación práctica, buscando que el estudiante desarrolle capacidades de trabajo grupal, y le sea posible establecer los pasos a seguir para la resolución de situaciones problema.

En última instancia, se verificará a través de la autonomía desarrollada por el sujeto pedagógico, la cual se logra mediante la adquisición y comprensión progresiva de los diversos contenidos, tanto horizontal como verticalmente dispuestos

CARGA HORARIA

2 Horas Semanales
72 Horas Anuales

Finalizado el cursado de la materia los y las estudiantes deberán haber desarrollado las siguientes Capacidades:

- Manejo de distintos sistemas de numeración.
- Comprensión de circuitos y su funcionamiento.
- Interpretar enunciados.
- Identificar datos y resultados.
- identificar sistemas de numeración, realizar pasajes.
- Resolver problemas en el marco de las diferentes aplicaciones.
- Reconocer y resolver circuitos, diagramas circuitales y compuertas.
- Resolver problemas simples mediante diagramación lógica.
- Seleccionar la herramienta adecuada según la plataforma sobre la cual se realizará la solución.
- Utilizar maxiterminos, miniterminos, simplificación.
- Elaborar documentación pertinente.
- Implementar y evaluar la solución desarrollada.

**CAPACIDADES A
DESARROLLAR**

CONTENIDOS	<p>Sistemas de numeración. Base de un sistema. Distintas bases: sistema decimal, binario, octal y hexadecimal. Cambios de base. Operaciones aritméticas. Funciones lógicas, representación, tabla de estados. Álgebra de Boole, postulados, propiedades y, teoremas Circuitos combinacionales. Compuertas. Diagrama Circuitual. Implementación de funciones lógicas a través de circuitos de compuertas .Formas canónicas de una función, miniterminos y maxiterminos, adyacencias, simplificación. Funciones aritméticas, números con formato, suma de números signados. Complemento a la base y a la base menos uno como forma de obtener números signados.</p> <p>Distancia detección y corrección de errores, códigos con redundancia, paridad, hamming.</p> <p>Códigos ASCII.</p> <p>Estándares IEEE para representación de números de punto flotante.</p> <p>Lenguajes aptos para trabajo a bajo nivel: lenguaje C y su relación con bibliotecas. Uso en ejemplos prácticos.</p>
ORIENTACIONES DIDÁCTICAS	<ul style="list-style-type: none"> - Aprendizaje colaborativo. - Uso y análisis de la información. - Contacto con la realidad a través de proyectos técnicos - Aprendizaje basado en problemas - Técnica del debate - Programas de Simulación. - Técnica de la pregunta - Autoaprendizaje. Investigación - Búsqueda y análisis de información - Métodos de consenso. Debates.

5° AÑO FORMACIÓN CIENTÍFICO TECNOLÓGICA	
BASES DE DATOS	
CARGA HORARIA	2 Horas Semanales 72 Horas Anuales
CAPACIDADES A DESARROLLAR	Finalizado el cursado de la materia los y las estudiantes deberán haber desarrollado las siguientes Capacidades: <ul style="list-style-type: none"> - Construcción de una pequeña base de datos. - Comprensión de tipos de datos. - Identificar posibles problemas y diseñar estrategias de resolución mediante algoritmos.
CONTENIDOS	
<p>Conceptos generales en la gestión de datos. DBMS. Comparación con los archivos tradicionales. Dependencias funcionales. Reglas de derivación. Cobertura mínima. Normalización. Modelo entidad– relación. Álgebra relacional. Modelo relacional. Diseño de BD, tablas, consultas, informes, macros, procedimientos almacenados, ítem, triggers. Lenguajes de un DBMS. SQL, QBE. Técnicas de implementación y administración de bases de datos. Visualización, seguridad y autorización en BD.</p> <p>Proyecto de aplicación de sistema de bases de datos. Utilización de lenguaje SQL. Creación de una Base de Datos. Definición de campos. Tipos de datos. Grabación de la Base de Datos creada. Generación de Tablas. Recuperación y Modificación de la base de datos creada. Modificación del tipo y cantidad de campos. Carga de datos en la Base. Agregado y edición de registros. Posicionamiento en la base. Manejo de Datos. Listados y Reportes, características generales. Visualización de registros: individuales, condicionales, por grupos. Consultas de tablas individuales e interrelacionadas.</p>	

ORIENTACIONES DIDÁCTICAS

- Uso de DBMS.
- Conceptos básicos de cardinalidad y normalización
- Aprendizaje colaborativo.
- Uso y análisis de la información.
- Contacto con la realidad a través de proyectos técnicos
- Aprendizaje basado en problemas
- Técnica del debate
- Programas de Simulación.
- Técnica de la pregunta

<div>5° AÑO</div> <div>FORMACIÓN CIENTÍFICO TECNOLÓGICA</div> <div>MODELOS Y SISTEMAS I</div>	
CARGA HORARIA	<div>3 Horas Semanales</div> <div>108 Horas Anuales</div>
CAPACIDADES A DESARROLLAR	<p>Finalizado el cursado de la materia los y las estudiantes deberán haber desarrollado las siguientes Capacidades:</p> <ul style="list-style-type: none"> - Resolver situaciones problemáticas en el marco de las diferentes aplicaciones. - Interpretar consignas. - Resolver algoritmos en los que se apliquen estructuras condicionales. - Identificar métodos, para su correcta aplicación en diferentes situaciones (tomadas de la realidad). - Resolver problemas simples mediante diagramación lógica. - Seleccionar la herramienta adecuada según la plataforma sobre la cual se realizará la solución.
CONTENIDOS	<p>Herramientas de Diagramas: de flujo de datos, de control. Diccionario de datos. Almacenamiento de datos. Convenciones. Diagrama de transición de estado. Diagrama de entidad. Relación. Técnicas para el Análisis: Desarrollo control y seguimiento del proyecto. Prototipo. Ciclo de vida. Métodos de asignación y transporte: Método húngaro. Aproximación por costos mínimos. Soluciones alternativas para cada caso. Otros métodos. Números al Azar: Funciones generadoras de números aleatorios. Métodos, manuales y algoritmos. Comparación. Teoría de los modelos: Tabla y árboles de decisión. Abstracción de datos. Objetivos en el uso de modelos matemáticos. Clasificación de modelos. Predicción de los resultados. Modelos de experimentación controlados. Diseño de documentación y manuales de sistemas y usuarios: Técnicas para la elaboración de</p>

la documentación de las tareas. Lineamientos generales para la confección del manual. Conformación de la organización de la documentación. Grafos y redes: Problemas básicos. Transportes. Flujo máximo. Ruta más corta. Camino mínimo. Camino crítico. Algoritmos básicos.

ORIENTACIONES DIDÁCTICAS

- Generar consignas que incluyan conocimientos previos de otras materias, que puedan utilizarse como base de aplicación en consignas propias de esta materia.
- Poner en evidencia la relación teoría y práctica, con la utilización de anotaciones en carpeta, para consulta de las resoluciones.
- Leer y analizar situaciones problemáticas, para luego resolver paso a paso.
- Utilizar material multimedia como recurso didáctico para presentación de temática en curso.
- Utilizar registros como parte del análisis de las consignas, para una correcta interpretación.

<div> <div>5° AÑO</div> <div>FORMACIÓN TÉCNICA ESPECÍFICA</div> </div> <div> <div>LENGUAJES DE PROGRAMACIÓN I</div> </div>	
<div>CARGA HORARIA</div>	<div> <div>4 Horas Semanales</div> <div>144 Horas Anuales</div> </div>
<div> <div>CAPACIDADES A DESARROLLAR</div> </div>	<div> <div>Finalizado el cursado de la materia los y las estudiantes deberán haber desarrollado las siguientes Capacidades:</div> <ul style="list-style-type: none"> - Establecer diferencia y flexibilidad ente ambos lenguajes de programación orientados a objetos. - Diseñar estrategias de resolución de forma autónoma. - Resolver problemas complejos mediante diagramación lógica. - Elaborar documentación pertinente. - Trabajo colaborativo entre pares. - Implementar y evaluar la solución desarrollada. </div>
<div>CONTENIDOS</div>	<div> <div>Uso simple de registros para representar información de entidades. Listas: un tipo de datos de tamaño dinámico y no acotado. Algoritmos sobre listas: inserción al inicio, al final y en la enésima posición. Funciones sencillas de tipo acumulación (maps, filters y zips concretos).</div> <div>Algoritmos sobre lista con una propiedad invariante: inserción en una ubicación relativa a los elementos contenidos, funciones sencillas que hagan uso de la propiedad. Manipulación de cadenas de caracteres. Diferencias y similitudes con listas. Reutilización de los conceptos de listas. Pasaje de parámetros por copia y por referencia. Arreglos como tipo estático: declaración, indexación, alteración de una posición. Pasaje de arreglos por parámetro, modificación y uso como tipo de retorno dentro de una función. Iteración sobre arreglos.</div> <div>Algoritmos básicos para buscar, ordenar, mezclar y realizar el producto cartesiano de dos o más conjuntos</div> </div>

ORIENTACIONES DIDÁCTICAS

Se dará continuación pedagógica a los contenidos desarrollados en el año anterior, para luego introducir a los estudiantes a la programación C/C++.

- Esto permitirá realizar un paralelismo entre ambos lenguajes de programación.
- Utilizar lenguajes orientados a funciones y métodos
- Se presentarán situaciones problemáticas que faciliten el trabajo colaborativo y el debate sobre estrategias resolutivas.
- Diseñar y crear algoritmos de manera autónoma.
- Aplicar los contenidos en contextos reales.
- Controlar, verificar y reformular la implementación de los programas realizados.

5° AÑO FORMACIÓN TÉCNICA ESPECÍFICA	
REDES INFORMÁTICAS	
CARGA HORARIA	4 Horas Semanales 144 Horas Anuales
CAPACIDADES A DESARROLLAR	<p>Finalizado el cursado de la materia los y las estudiantes deberán haber desarrollado las siguientes Capacidades:</p> <ul style="list-style-type: none"> - Que el alumno adquiera conocimientos sobre los diferentes tipos de redes. - Clasificar las redes según su alcance y topología. - Armar estructuras y cables según los diferentes protocolos y normas. - Interpretar y realizar un mapa de red utilizando las normativas vigentes. - Identificar los diferentes tipos de servidores. - Diferenciar diferentes dispositivos como routers, switch y hub y configurarlos según corresponda
CONTENIDOS	
<p>Concepto de redes. Cobertura de las redes. Interoperatividad. Razones para instalar una red de computadoras. Clasificación, WAN, MAN, LAN, Redes Cliente Servidor, Peer to Peer. LAN, topología física y lógica. Especificaciones LAN, Ethernet, IEEE 802.x , FDDI. Modelo ISO/OSI, capas, descripción, funciones. subcapas . Análisis de las capas de medios. Modelo TCP/IP, comparación con el modelo OSI, Protocolo TCP y UDP, Protocolo IP, numeración IP v4 y v6, compatibilidad. Placa de red, dirección MAC, unicast. multicast y broadcast. Encapsulación, MTU. Cableado, cable UTP, conectorización, Normas EIA/TIA 568B. Dispositivos de red, hub, switch, router. Características, limitaciones. Dominio de colisión, dominio de broadcast. Segmentación. Ethernet 100BaseTX, Gigabit Ethernet. Switch1ng, métodos, store and forward, cut-through y fragment free, operación a N2, redundancia, Spanning Tree Protocol. Configuración de switches, modo usuario y modo privilegiado, comandos. Nociones de Cableado</p>	

Estructurado, Cuarto de comunicaciones, Racks. Cablearlo horizontal, elementos que lo componen. Análisis del tráfico en una LAN. Sniffing analizadores de protocolo, captura del tráfico, filtros de captura, análisis. Redes inalámbricas, principios de funcionamiento, protocolos, modos ad-hoc e infraestructura, usos, limitaciones seguridad, distintos tipos de antena y su aplicación

ORIENTACIONES DIDÁCTICAS

- Uso de simuladores, al estilo Packet tracer- visual router-sniffer monitor.
- Practicas con enrutadores.
- Uso de geolocalizadores
- Se presentarán situaciones problemáticas que faciliten el trabajo colaborativo y el debate sobre estrategias resolutivas.
- Diseñar y crear algoritmos de manera autónoma.
- Aplicar los contenidos en contextos reales.

5° AÑO FORMACIÓN TÉCNICA ESPECÍFICA	
DISEÑO WEB	
CARGA HORARIA	2 Horas Semanales 72 Horas Anuales
CAPACIDADES A DESARROLLAR	Finalizado el cursado de la materia los y las estudiantes deberán haber desarrollado las siguientes Capacidades: <ul style="list-style-type: none"> - Interpretar la composición de páginas web de todo tipo. - Maquetado en diferentes software y utilización de herramientas y filtros para el armado completo. - Conocer y trabajar con diferentes tipos de archivos (gráficos – audiovisuales – etc.)
CONTENIDOS	
Manejo de Imágenes. Tipo de imágenes. Conversión de tipos. Resolución y tamaño. Retoques y filtros. Composición de imágenes. Software para animación. Herramientas de dibujo. Selección y modificación de objetos. Texto. Creación y transformación de objetos. Animación en una y en múltiples escenas. Línea de tiempo. Tipos de fotogramas. Previsualización de la acción. Capas: propiedades de la capa, capas múltiples y capa máscara. Símbolos: gráfico, clip de película y botón. Importar objetos. Biblioteca de símbolos. Animación fotograma a fotograma. Interpolación de movimiento y forma. Propiedades de la animación. Animación por trazado: añadir capa guía de movimiento. Animaciones de máscara. Exportación de películas. Añadir sonidos a las animaciones y botones. Propiedades del sonido. Acciones básicas. Software de diseño Web. Características del Programa. Configuración del sitio y de los archivos del proyecto. Creación de un diseño de página basado en tablas. Inserción de tablas. Propiedades de una tabla. Marcadores. Adición de contenido a las páginas. Inserción de imágenes, animaciones y videos. Inserción de texto. Creación de vínculos. Vista previa de la página en un navegador. Creación de hojas de estilo CSS. Adjuntar hoja de estilos. Diseño por capas.	

ORIENTACIONES DIDÁCTICAS

- Armado de página simple utilizando todos los recursos.
- Trabajo práctico orientado a las diferentes propuestas planteadas.
- Se presentarán situaciones problemáticas que faciliten el trabajo colaborativo y el debate sobre estrategias resolutorias.
- Diseñar y crear algoritmos de manera autónoma.
- Aplicar los contenidos en contextos reales.

5° AÑO FORMACIÓN TÉCNICA ESPECÍFICA	
ARQUITECTURA DE BASE DE DATOS	
CARGA HORARIA	2 Horas Semanales 72 Horas Anuales
CAPACIDADES A DESARROLLAR	<p>Finalizado el cursado de la materia los y las estudiantes deberán haber desarrollado las siguientes Capacidades:</p> <ul style="list-style-type: none"> - Selección del software gestor de base de datos que se adapte a las necesidades del contexto. - Análisis de funcionamiento de las organizaciones para la recolección de datos a utilizar en el diseño de base de datos. - Realizar el uso de una herramienta de desarrollo de base de datos - Diseñar los diagramas de entidad relación - Instalar y configurar un servidor de base de datos - Aplicar la normalización de datos que se ajuste a las necesidades del problema en cuestión. - Realizar consultas: con parámetros, de selección, de actualización, etc. - Implementar y seguir el funcionamiento de la base de datos
CONTENIDOS	
<p>Objetivos de las bases de datos. Usos y aplicaciones de las bases de datos. Definición de modelo de datos. Modelo entidad relación. Entidades, atributos y relaciones. Llaves. Cardinalidad de las entidades en una relación. Dependencia de existencia y de identificación. Generalización y especialización. Agregación. Consideraciones de diseño. Normalización. Dependencias funcionales. Primeras formas normales. 1FN. 2FN. 3FN y FNBC(forma normal Boyce-Codd. Normalización adicional. Dependencia multivaluada y 4FN. Dependencia de junta y 5FN. Integridad de bases de datos. Concepto. Restricciones básicas (not null, llave primaria, orden, verificación y aserción). Integridad de entidad. Integridad referencial. Reglas de</p>	

base de datos. Reglas de negocios. Recuperación de bases de datos. Transacciones. Definición de transacción. Bitácora. Tipos de bitácora. Contenido de la bitácora. Diccionario de datos. Concepto. Contenido y función. Tipos. Álgebra relacional. Cálculo relacional. Pasos para implementar un gestor de DB, Diseño de BD, tablas, consultas, informes, macros, procedimientos almacenados. Sentencia create, ligaduras, eliminación de tablas, sentencia alter, consultas simples, sentencia select– from, columnas calculadas, condiciones de búsqueda, ordenación de resultados (order by) , consultas a múltiples tablas (inner join). Las operaciones de Actualización, Supresión e Inserción. Técnicas de implementación y administración de bases de datos. Visualización, seguridad y autorización en BD

ORIENTACIONES DIDÁCTICAS

- Exponer los contenidos con analogías y aplicaciones de sistemas de bases de datos a la realidad
- Seleccionar el gestor de base de datos a utilizar de acuerdo a las necesidades del entorno, el contenido del programa y los recursos disponibles.
- Utilizar el gestor de base de datos seleccionado en el desarrollo de las prácticas de laboratorio.
- Involucrar al estudiante en el uso de herramientas de software para el diseño de bases de datos.
- Elaborar en un conjunto de ejercicios asociados al entorno, para que el estudiante diseñe la base de datos utilizando algún modelo.
- Posicionar al alumno en situaciones problemáticas y permitir que determine las posibles variables
- Fomenta en los estudiantes la conformación de equipos de trabajo para el desarrollo de proyectos de su interés, aplicando los conocimientos adquiridos durante el curso.
- Involucrar al estudiante en las innovaciones generadas en modelos, gestores y lenguajes de base de datos, propiciando que éste se documente en distintas fuentes de información y sean analizadas en clase.
- Definir un esquema de documentación de los trabajos realizados o desarrollo de aplicaciones, que contenga las estrategias metodológicas aplicadas según sea el caso.

<div> <div>6° AÑO</div> <div>FORMACIÓN CIENTÍFICO TECNOLÓGICA</div> </div> <div> <div>MATEMÁTICA APLICADA</div> </div>	
<div>CARGA HORARIA</div>	<div> <div>3 Horas Semanales</div> <div>108 Horas Anuales</div> </div>
<div>CAPACIDADES A DESARROLLAR</div>	<div> <div>Finalizado el cursado de la materia los y las estudiantes deberán haber desarrollado las siguientes Capacidades:</div> <ul style="list-style-type: none"> - Establecer transferencias pertinentes de los conceptos matemáticos a situaciones de la especialidad. - Comprobar la importancia de la formalización como herramienta de comunicación en el ámbito de la Matemática. - Utilizar software de aplicación en modelos matemáticos. - Interpretar conceptos de probabilidad y estadística. - Aplicar de manera autónoma diversas estrategias en la resolución de problemas. </div>
	<div>CONTENIDOS</div>
	<div>Combinatoria. Técnicas de conteo. Variaciones, permutaciones. Combinaciones. Grafos, árboles, diégrafos y multigrafos. Construcción de grafos, isomorfismos de grafos. Representación de grafos matriz de adyacencias, matriz de incidencias. Caminos, ciclos y grafos conexos. Árboles de búsqueda binarios. Árboles de decisión.</div>

ORIENTACIONES DIDÁCTICAS

- Resolución de situaciones problemáticas vinculadas a la especialidad.
- la evaluación escrita.
- trabajo en grupo.
- utilización de bibliografía específica.

6° AÑO
FORMACIÓN
CIENTÍFICO
TECNOLÓGICA

SISTEMAS DIGITALES II

CARGA HORARIA
2 Horas Semanales
72 Horas Anuales

- Comprensión de circuitos y su funcionamiento.
- Interpretar enunciados.
- Identificar datos y resultados..
- Resolver problemas en el marco de las diferentes aplicaciones.
- Reconocer y resolver circuitos, diagramas circuitales y compuertas.
- Resolver problemas simples mediante diagramación lógica.
- Seleccionar la herramienta adecuada según la plataforma sobre la cual se realizará la solución.
- Elaborar documentación pertinente.
- Implementar y evaluar la solución desarrollada.

**CAPACIDADES A
DESARROLLAR**

CONTENIDOS

Familias y subfamilias lógicas. Clasificación, características, comparación. Circuitos secuenciales. Conceptos de memoria de un bit. Flip Flops. Contadores y Registros. Lógica secuencial. Concepto de realimentación en un sistema lógico combinacional. Circuitos biestables: R-S asíncrono. Tabla de verdad. Ecuación característica. Condición de restricción. Biestable como elemento básico de memoria. Biestables síncronos: R-S, J-K, D y T. Tabla de verdad. Ecuación característica. Diagramas temporales. Entradas asíncronas o de fuerza. Función. Utilidad. Biestable Master/Slave. Implementación de circuitos utilizando los diferentes tipos de biestables. Registros. Implementación de registros utilizando biestables. Restricciones.

Registros de desplazamiento. Tipos de registros: serie/serie, serie/paralelo, paralelo/serie, paralelo/paralelo. Diagramas temporales asociados. Concepto de memoria de “n” bits. Contadores. Características de los circuitos contadores: tipo, módulo, secuencia. Implementación de contadores utilizando bistables.

Restricciones. Contadores en anillo: memoria circulante. Implementación. Diagramas temporales. Contadores asíncronos: características. Síntesis de contadores asíncronos progresivos y regresivos.

Diagramas temporales. Contadores síncronos: características. Síntesis de contadores síncronos progresivos y regresivos de cualquier módulo.

Diagramas temporales. Confección de la tabla de transiciones de estado de entradas y salidas según el biestable utilizado. Memorias, Conceptos básicos.

Tipos de memoria. Clasificación. Diferentes tecnologías. Campo de utilización de cada tipo. Bancos de memoria. Implementación y organización.

Asignación de las líneas necesarias en los buses de dirección y de datos.

Identificación de los bloques de memoria utilizando notación binaria y hexadecimal.

Introducción a La Arquitectura de Computadoras. Organización básica de un CPU – Registros y Buses – Clocks – Subsistema de Entrada / Salida – Organización y direccionamiento de la memoria. Interrupciones. Concepto de microprocesador y microcontrolador. Ejemplos.

Arquitectura de Computadoras. Arquitectura de un microprocesador. Circuitos LSI / VLSI. Evolución histórica de los sistemas computacionales en relación al avance científico tecnológico de los microprocesadores. Registros y buses. Ciclos de lectura y escritura. Tiempos de acceso. Áreas de datos, código y pila. Interrupciones enmascarables y no enmascarables. Interrupciones por hardware y por software. El modelo de programación. Nociones de direccionamiento de memoria en modo real y modo protegido. Nociones de paginación de memoria. Modos de direccionamiento. Concepto de DMA. Direcciones de periféricos: Ports de E/S. Estructuras de interconexión. Interconexión con buses. Concepto de Instrucción. Código de operación. Operandos. Tipo de instrucciones. Ejecución de instrucciones por parte del CPU. Ciclos de instrucción, máquina y reloj. Ejecución de programas básicos. Introducción al lenguaje Assembler x86. Concepto de Firmware. ALU. Operaciones aritméticas y lógicas. Registro de estado.

Introducción al entorno de Unity. Conceptos básicos de Unity. Introducción al motor de juegos

ORIENTACIONES DIDÁCTICAS

- Utilización de plataformas móviles, soportes digitales, software de diferentes sistemas.
- Ubicación en gráfica y presentación visual.
- Aprendizaje colaborativo.
- Uso y análisis de la información

<div>6° AÑO</div> <div>FORMACIÓN CIENTÍFICO TECNOLÓGICA</div> <div>SISTEMAS DE GESTIÓN Y AUTOGESTIÓN</div>	
CARGA HORARIA	<div>3 Horas Semanales</div> <div>108 Horas Anuales</div>
CAPACIDADES A DESARROLLAR	<p>Finalizado el cursado de la materia los y las estudiantes deberán haber desarrollado las siguientes Capacidades:</p> <ul style="list-style-type: none"> - Reconocer los vacíos legales que surgen en determinados marcos jurídicos. - Diseñar diferentes sistemas de gestión - Diseñar estrategias en el resguardo de información teniendo en cuenta su carácter público y privado. - Desarrollar estrategias actitudinales en los debates, formulación de posturas y confrontaciones priorizando el diálogo y respeto por el otro.
CONTENIDOS	
<p>Mercado informático: Conceptos básico de micro y macro economía. CESSI (Cámara de empresas de servicios y sistemas informáticos). OSI (Observatorio de evolución del mercado informático en Argentina). Estado consumidor y proteccionista en el mercado informático. Marco jurídico legal del mercado informático.</p> <p>Interrelación de las principales variables económicas y las políticas económicas actuales. Principales empresa multinacionales en nuestro país. Desarrollo de la tecnología informática en nuestro país (hardware y software). Posicionamiento de la Argentina en el mercado informático mundial. Evolución de la organización social y su impacto en las instituciones. Poder, liderazgo y autoridad: interpretación de estos conceptos en el juego de roles.</p> <p>Diferentes enfoque cronológico de la organización: mecanicista, científico, rrhh, neoclásico, burocrático, APO, sistémico, otros.</p>	

Deontología profesional: códigos de ética profesional. Organización como sistema social, concepto, fines y funciones.

Autogestión: psico-sociología de las organizaciones. Comunicación, conflictos y resolución de conflictos.

El proceso de toma de decisiones, el papel de la información en el mismo. Principales normas jurídicas e impositivas, o prácticas profesionales o comerciales que impliquen la necesidad de preservar o resguardar datos

ORIENTACIONES DIDÁCTICAS

- Investigaciones y lectura de diferente autores sobre sociología y cómo evolucionó la organización de los estados y sociedades después de la revolución francesa.
- Debatir sobre las diferentes posturas de los medios de comunicación referentes a las variables económicas, la realidad y el marco teórico. Realizar comparaciones sobre la evolución histórica de las organizaciones y que rasgos se observan aún en las organizaciones actuales.
- Lectura de leyes y artículos que estén relacionados con la actividad informática tanto en su desarrollo como en el resguardo de información.
- Reproducir textos argumentativos, informativos y expositivos de los diferentes contenidos.
- Interpretaciones de parámetros, tendencia y gráficos estadísticos.

<div> <div>6° AÑO</div> <div>FORMACIÓN CIENTÍFICO TECNOLÓGICA</div> </div> <div> <div>SEGURIDAD INFORMÁTICA</div> </div>	
<div>CARGA HORARIA</div>	<div> <div>3 Horas Semanales</div> <div>108 Horas Anuales</div> </div>
<div> <div>CAPACIDADES A DESARROLLAR</div> </div>	<p>Finalizado el cursado de la materia los y las estudiantes deberán haber desarrollado las siguientes Capacidades:</p> <ul style="list-style-type: none"> - Acercar al alumno a una realidad más amplia en la que debe obrar la seguridad informática, a través de múltiples contextos de nivel del negocio. - Conocer e interpretar la normativa vigente en cuanto a seguridad y el desempeño del profesional informático. - Generar algoritmos propios de encriptación tomando las bases de la criptografía clásica. - Conocer un propósito distinto de los lenguajes de programación a los adquiridos en otras áreas. - Adquirir nociones básicas de informática forense.
<div>CONTENIDOS</div>	<p>El impacto de las computadoras en la sociedad. La función de las computadoras en los negocios, la gestión y el proceso global de toma de decisiones. Responsabilidades éticas y legales de los profesionales de la informática. Seguridad, introducción. Seguridad Informática, definición. Recursos y vulnerabilidades. Finalidad y Estrategia de la Seguridad Informática.</p> <p>Revisión de normativa vigente respecto del área de seguridad informática. Nacionales e Internacionales. Áreas en las que usualmente recae la responsabilidad de Seguridad, inconvenientes respecto a su control.</p> <p>Seguridad Física. Selección y diseño, metodología de evaluación. Protección de acceso. Medidas de resguardo de almacenamiento. Riesgos, distintos tipos, evaluación de ocurrencia.</p>

Seguridad Lógica. Concepto. Riesgos de Seguridad y problemas de protección. Recursos a proteger. Metodologías de uso común: identificación y autenticación de usuarios. Redes. LAN, WAN, de teleproceso. Componentes. Criptosistemas, propiedades. Interfaces físicas y lógicas. Nociones de Criptosistemas en redes. Algoritmos clásicos de encriptación. Implementación en lenguajes de programación. Informática forense. Nociones básicas.

ORIENTACIONES DIDÁCTICAS

- Generar en el aula un ambiente de colaboración, respeto y entusiasmo por aprender.
- Ofrecer ejemplos variados y suficientes, de casos reales y actuales.
- Proporcionar el marco normativo actualizado, relevante a la asignatura.
- Ayudar a los estudiantes a reconocer contextos con los que aún no estén familiarizados.
- Recordar nociones básicas de algoritmos, si fuere oportuno

6° AÑO FORMACIÓN CIENTÍFICO TECNOLÓGICA		DERECHOS DEL TRABAJO	
CARGA HORARIA		2 Horas Semanales 72 Horas Anuales	
CAPACIDADES A DESARROLLAR		Finalizado el cursado de la materia los y las estudiantes deberán haber desarrollado las siguientes Capacidades: <ul style="list-style-type: none"> - Conocer el derecho laboral como una rama del derecho privado y reconocer los principios que lo fundamentan. - Reconocer los derechos y deberes de los trabajadores y el empleador. - Interpretar la ley 20.744 y reconocerla como la principal fuente del derecho del trabajo. - Interpretar situaciones y poder emitir juicios ante eventuales conflictos. 	
CONTENIDOS		Derecho de trabajo. Tipos de trabajadores. Derechos y obligaciones de las partes. Principios del derecho. Contrato de Trabajo. Períodos de prueba. Remuneración. Recibo de haberes. Sueldo mínimo vital y móvil. Estabilidad Laboral. Aportes y Contribuciones. Recibo de haberes. Jornada de trabajo. Descanso semanal. Horas suplementarias. Feriados Nacionales. Días no laborables. Licencias especiales. Vacaciones. ART. (Ley 2455). Enfermedades y Accidentes de trabajo. Asignaciones familiares. Liquidación de haberes. Planillas del empleado y empleador. Exigibilidad de derechos. Mecanismos y organismos de exigibilidad de derechos laborales. Ética en el desempeño profesional. Trabajo decente. PyMES. Empresas recuperadas. Micro emprendimientos. Microeconomía.	

ORIENTACIONES DIDÁCTICAS

- Métodos de consenso. Charlas. Debates. Discusión en Panel. Seminario
- Cuadros comparativos.
- Aprendizaje colaborativo.
- Uso y análisis de la información.
- Contacto con la realidad del mercado laboral
- Investigaciones
- Análisis de casos.
- Presentaciones Digitales

<div> <div>6° AÑO</div> <div>FORMACIÓN TÉCNICA ESPECÍFICA</div> </div> <div> <div>LENGUAJES DE PROGRAMACIÓN II</div> </div>	
<div>CARGA HORARIA</div>	<div> <div>4 Horas Semanales</div> <div>144 Horas Anuales</div> </div>
<div>CAPACIDADES A DESARROLLAR</div>	<div> <div>Finalizado el cursado de la materia los y las estudiantes deberán haber desarrollado las siguientes Capacidades:</div> <ul style="list-style-type: none"> - Resolver problemas complejos entendiendo las estructuras de datos y avanzando en los saberes previos a través de la programación orientada a objetos y bases de datos - Seleccionar las herramientas adecuadas. - Elaborar documentación pertinente. - Implementar y evaluar la solución desarrollada. </div>
	<div>CONTENIDOS</div>
	<div> <p>Teoría de objetos: Origen de la programación orientada a objetos. Clases vs. Objetos. Ciclo de vida de los objetos: constructores y destructores. Propiedades de un objeto. Calificadores de las propiedades. Métodos de acceso. Calificadores de los métodos. Nivel de acceso de métodos y propiedades. Variables de clase vs variables de instancia. Mensajes entre objetos. Relaciones entre objetos. Conceptos de herencia. Herencia simple. Herencia múltiple. Superclases y subclases. Interfaces. Clases abstractas. Polimorfismo. Sobrecarga de métodos. Sobrecarga de operadores. Software orientado a componentes y frameworks. Lenguaje de Programación: Introducción al lenguaje y .NET. El framework, .NET. Arquitectura básica de .NET: Entidades y Componentes. Biblioteca de clases: Espacios de nombre. La directiva using. Ensamblados. Características de un proyecto: componentes básicos. Modo Consola vs. Interface gráfica. Sintaxis básica del lenguaje. Definición y creación de Clases y objetos. Atributos y métodos. Modificadores de acceso. Tipos de datos. Operadores aritméticos, relacionales y lógicos. Conversiones de Tipos de datos. Manejo de</p> </div>

excepciones: try catch finally. Estructuras de control: Condicionales y de repetición. Arreglos de distintas dimensiones. Arreglos de objetos. Los arreglos como atributos de clases. Accesos por referencia y valor. Formularios: características. Interfaz de usuario en .NET. Controles: Propiedades y eventos. Componentes y dll. Los eventos Combinar formularios. Paneles. Arboles. Arreglos de controles. El espacio System.Drawing. Gráficos e Impresión en .NET. Los gráficos básicos. Construir graficas estadísticas sobre nuestros datos. La impresión. Control de la impresión. Aplicaciones de impresión. Manejo de Base de datos: Bases de datos relacionales. Manejadores de Bases de datos, los motores. Relevamiento de datos. Normalización. Validaciones de datos. Lenguaje SQL. Creación y manejo de Bases de datos. Creación y manejo de tablas. El manejo de los datos propiamente dichos. Consultas de los datos almacenados. Acceso a Datos desde una Aplicación, ADO .NET. La importancia de los datos en las aplicaciones. La capa de acceso a datos. Arquitectura de ADO .NET. Clases de ADO .NET. Construcción de ABM. Consultas con ADO .NET.

ORIENTACIONES DIDÁCTICAS

- Resolución de problemas ya establecidos y desarrollo de proyectos en conjunto con los alumnos.
- Se presentarán situaciones problemáticas que faciliten el trabajo colaborativo y el debate sobre estrategias resolutorias.
- Diseñar y crear algoritmos de manera autónoma.
- Aplicar los contenidos en contextos reales.

6° AÑO		
FORMACIÓN TÉCNICA ESPECÍFICA		
PROGRAMACIÓN Y CONTROLES AUTOMATIZADOS		
CARGA HORARIA	2 Horas Semanales	
	72 Horas Anuales	
	Finalizado el cursado de la materia los y las estudiantes deberán haber desarrollado las siguientes Capacidades:	
CAPACIDADES A DESARROLLAR	<ul style="list-style-type: none">- Resolver problemas.- Interpretar enunciados.- Identificar datos y resultados.- Seleccionar las herramientas adecuadas.- Elaborar documentación pertinente.- Implementar y evaluar la solución desarrollada.- Implementar lenguajes de bajo nivel y lenguaje de control	
	CONTENIDOS	
Introducción a la Arquitectura de microcontroladores. Diferencia entre microprocesador y microcontrolador, modelo de Von Neuman y Modelo Harvard. Memoria de programa. Memoria de Datos. Descripción de los registros del microcontrolador. Descripción de las líneas de entrada/salida, entradas/salidas dedicadas. Líneas de interrupción. Pila, concepto de subrutina. Repertorio de instrucciones del microcontrolador. Programación en lenguaje assembler, diagramas de flujo, desarrollo de programas básicos en lenguaje assembler. Noción de programa Compilador, directivas básicas, noción de utilización de programa Simulador, noción de programa Programador. Noción de PLC. Utilización, características, entradas/salidas. Programación básica en lenguaje Ladder, Elementos de la programación: variables, contactos, bobinas, words, bits del sistema, flags sin retención,		

flags con retención, bloques operación, bloques comparación, bloque temporizador (timer), bloque monoestable, bloque contador, bloque programador cíclico, bloque PID, bloque encoder. Edición de un programa ladder: ubicación de símbolos gráficos, contactos, bobinas, bloques timer, monoestables, etc., bloques operación y comparación.

ARDUINO - Sensores varios (Sensor De Nivel De Agua Y Liquidos Arduino , Sensor Tension Voltaje 0v-25v Dc Arduino,Sensor Optico Reflectivo Infrarrojo,Sensor Digital De Temperatura ,Sensor Humedad Relativa Y Temperatura, Sensor De Agua Lluvia Meteorologia, Detector De Fuego Sensor De Llama, Sensor Velocidad Giro Rueda Pulsos Tacometro, Sensor De Luz Con Ldr

Fotoresistor, Sensor Obstaculos Infrarrojo, Sensor Infrarojo ,Sensor Humedad Relativa Y Temperatura, Sensor Movimiento Infrarrojo, Sensor De Sonido Microfono Regulable,Sensor Ultrasonido Hc-sr04 Distancia. Programacion de Arduino y sensores.

Actuadores (Motores paso a paso -PAP-, de Corriente continua y alterna CC y CA), Lámparas de incandescencia, Leds, Timbres o alarmas, Pulsadores, Interruptores, etc.

Circuitos Integrados

ORIENTACIONES DIDÁCTICAS

- Proyecto integrador: utilización de los sensores, motores, arquitectura y programación de dichos componentes
- Se presentarán situaciones problemáticas que faciliten el trabajo colaborativo y el debate sobre estrategias resolutivas.
- Diseñar y crear algoritmos de manera autónoma.
- Aplicar los contenidos en contextos reales.

<div> <div>6° AÑO</div> <div>FORMACIÓN TÉCNICA ESPECÍFICA</div> </div> <div>DESARROLLO DE APLICACIONES WEB ESTÁTICAS</div>	
<div>CARGA HORARIA</div>	<div>2 Horas Semanales</div> <div>72 Horas Anuales</div>
<div>CAPACIDADES A DESARROLLAR</div>	<p>Finalizado el cursado de la materia los y las estudiantes deberán haber desarrollado las siguientes Capacidades:</p> <ul style="list-style-type: none"> - Descubrir nuevos espacios para el desarrollo de nuevas tecnologías. - Plantear un diseño web de acuerdo al tema a presentar. - Reconocer y utilizar el lenguaje de estilos de presentación (css) - Utilizar el lenguaje HTML. - Comprender la eficacia en la utilización de un framework como JQuery. - Identificar los lenguajes cliente-servidor. - Analizar y resolver diferentes situaciones problemáticas. - Diseñar y trabajar con formularios avanzados y bajo entornos mysql. - Creación de etiquetas y jerarquías. - Comprender e interpretar diferentes manuales y/o tutoriales
	<div>CONTENIDOS</div>
	<p>Planificación de un proyecto. Elección y análisis del tema. Determinación de finalidad u objetivos del proyecto. Estudio de viabilidad: estudio de mercado (cliente, target, competencia) y estudio técnico. Relevamiento de información y determinación del contenido. Estructuras, determinación gráfica de la estructura de un sitio. Conceptos de diseño, estilos, metáforas. Conceptos de accesibilidad y usabilidad. Inrtroducción al HTML. HTML y</p>

XHTML. Estructura de un documento HTML. Etiquetas y atributos generales. Cabecera de un documento HTML. Sintaxis de las etiquetas. Body. Estructuración del texto. Diseño y desarrollo de páginas WEB simples. Prueba, análisis y comparación del código con distintos navegadores WEB. Manipulación de fuentes en HTML. Etiquetas para el enlace a sitios WEB y archivos. Vínculos a imágenes, videos y sonido. Generación de listas. Estructuración y atributos de las tablas. Formularios, declaración y manipulación. Diseño por capas. Hojas de estilos en cascada (CSS). Importancia de la programación utilizando el HTML estricto. Selectores. Declaraciones múltiples. Agrupación de estilos. Herencia. Ventajas de escribir código utilizando CSS. Diseño y desarrollo de sitios WEB para determinados navegadores. Códigos CSS como facilitadores de mantenimiento de los sitios WEB. Registros y delegación del dominio. Servidores de nombres de dominio (DNS). Publicación del Sitio. Servidores de alojamiento. Métodos de transferencia de archivos. FTP. Administración del Sitio.

ORIENTACIONES DIDÁCTICAS

- Creación de diferentes aplicaciones utilizando los saberes aprendidos
- Diseño web de un sitio completo en ambos tipos de aplicaciones
- Se presentarán situaciones problemáticas que faciliten el trabajo colaborativo y el debate sobre estrategias resolutorias.
- Diseñar y crear algoritmos de manera autónoma.
- Aplicar los contenidos en contextos reales.

6° AÑO
FORMACIÓN
TÉCNICA
ESPECÍFICA

DESARROLLO DE APLICACIONES WEB DINÁMICAS

CARGA HORARIA

4 Horas Semanales
144 Horas Anuales

Finalizado el cursado de la materia los y las estudiantes deberán haber desarrollado las siguientes Capacidades:

- Que el alumno pueda desarrollar todos los conocimientos para armar app para todas clases de dispositivos.
- Diseñar según la necesidad pautaada en cada proyecto para captar y generar una app acorde a la necesidad planteada.
- Descubrir nuevos espacios para el desarrollo de nuevas tecnologías.
- Plantear un diseño web de acuerdo al tema a presentar.
- Reconocer y utilizar el lenguaje de estilos de presentación (css)
- Utilizar el lenguaje HTML.
- Comprender la eficacia en la utilización de un framework como JQuery.
- Identificar los lenguajes cliente-servidor.
- Analizar y resolver diferentes situaciones problemáticas.
- Definir y programar en scripts.
- Definir y programar una cookie.
- Diseñar y trabajar con formularios avanzados y bajo entornos mysql.
- Creación de etiquetas y jerarquías.
- Comprender e interpretar diferentes manuales y/o tutoriales

**CAPACIDADES A
DESARROLLAR**

CONTENIDOS

Planificación de un proyecto. Elección y análisis del tema. Determinación de finalidad u objetivos del proyecto. Estudio de viabilidad: estudio de mercado (cliente, target, competencia) y estudio técnico. Relevamiento de información y determinación del contenido. Estructuras, determinación gráfica de la estructura de un sitio. Conceptos de diseño, estilos, metáforas. Introducción a los lenguajes de programación de app, (app inventor, thunkable)

Bloque de diseño y bloque de programación, emuladores. Conceptos de accesibilidad y usabilidad. Como subir una app a playstore-gooplay. Registro, propiedad intelectual.

Cabecera de un documento HTML. Sintaxis de las etiquetas. Body. Estructuración del texto. Diseño y desarrollo de páginas WEB simples. Prueba, análisis y comparación del código con distintos navegadores WEB. Manipulación de fuentes en HTML. Etiquetas para el enlace a sitios WEB y archivos. Vínculos a imágenes, videos y sonido. Generación de listas. Estructuración y atributos de las tablas. Formularios, declaración y manipulación.

Introducción a los lenguajes de clientes. Definición de scripts. Programación de scripts. Tratamiento, definición y tipos de variables. Operadores aritméticos, relacionales y lógicos. Estructuras condicionales y de repetición. Objetos, funciones y métodos. Objetos y funciones predefinidas. Los objetos del navegador. Formularios avanzados. Concepto de cookie. Las cookies y los lenguajes de clientes. Lenguajes de servidor. Diferencias con lenguaje de cliente. Introducción al PHP. Declaración de variables, constantes y tipos de datos. Funciones básicas. Operadores aritméticos, relacionales y lógicos. Estructuras condicionales. Estructuras de repetición. Funciones de usuario, pasajes por valor y por referencia. Vectores y matrices. Manipulación de cadenas de caracteres. Entorno MySQL. Transferencia de datos entre PHP y MySQL. ActionScript para la Web. Introducción al lenguaje XML. Creación de etiquetas y jerarquías. Conexión con PHP y ActionScript.

ORIENTACIONES DIDÁCTICAS

- Creación de diferentes aplicaciones utilizando los saberes aprendidos
- Diseño web de un sitio completo en ambos tipos de aplicaciones
- Se presentarán situaciones problemáticas que faciliten el trabajo colaborativo y el debate sobre estrategias resolutorias.
- Diseñar y crear algoritmos de manera autónoma.
- Aplicar los contenidos en contextos reales.

7° AÑO FORMACIÓN CIENTÍFICO TECNOLÓGICA			EMPRENDIMIENTOS E INNOVACIÓN PRODUCTIVA		
CARGA HORARIA			2 Horas Semanales 72 Horas Anuales		
CAPACIDADES A DESARROLLAR			<p>Finalizado el cursado de la materia los y las estudiantes deberán haber desarrollado las siguientes Capacidades:</p> <ul style="list-style-type: none"> - Desarrollar en virtud de las necesidades del mercado emprendimientos productivos de acuerdo a su modalidad que permita adquirir experiencias en el ámbito laboral y en la construcción del auto empleo como medio de vida. - Analizar políticas públicas en torno a la especialidad. - Establecer lineamientos centrales en la formulación de emprendimientos productivos. - Diseñar emprendimientos productivos sustentables relacionadas con la especialidad. - Utilizar herramientas metodológicas en la gestión y promoción del desarrollo local y en emprendimientos productivos. - Analizar casos y experiencias. 		
CONTENIDOS			<p>Teorías del Emprendedorismo. Emprendedorismo social, cultural y tecnológica. Emprendedorismo y Desarrollo Local. Emprendimientos Familiares. Nociones de Derecho para Emprendedores. Finanzas para Emprendedores. Marketing. Calidad en la Gestión de emprendimientos. Técnicas de Comunicación. Actitud Emprendedora. Laboratorio de ideas y oportunidades. Planeamiento de emprendimientos sociales y culturales. Planeamiento de negocios para emprendedores.</p> <p>El análisis de casos y la evaluación de experiencias.</p>		

Innovación productiva. Innovación tecnológica.
Sustentabilidad de modelo innovadores y cuestiones ambientales

ORIENTACIONES DIDÁCTICAS

- Elección de emprendimiento sustentable. Presentación del producto y plan de comercialización.
- Análisis FODA.
- Identificación de la zona de desarrollo del emprendimiento.
- Estudio de la oferta y la demanda del mercado de desarrollo.
- Laboratorio de ideas y evaluación de oportunidades. Planificación de vía de comunicación y difusión mediante folletos, medios electrónicos.

<p>7° AÑO</p> <p>FORMACIÓN CIENTÍFICO TECNOLÓGICA</p> <p>EVALUACIÓN DE PROYECTOS</p>	
CARGA HORARIA	<p>2 Horas Semanales</p> <p>72 Horas Anuales</p>
<p>CAPACIDADES A DESARROLLAR</p>	<p>Finalizado el cursado de la materia los y las estudiantes deberán haber desarrollado las siguientes Capacidades:</p> <ul style="list-style-type: none"> - Realizar un proyecto y plasmar sus ideas bajo una normativa de presentación. - Identificar y clasificar los diferentes métodos de evaluación. - Comprender los conceptos de evaluación, tipos y usos de los mismos. - Conocer las diferentes estructuras organizacionales, objetivos y funciones de un departamento de informática.
CONTENIDOS	
<p>Normativa para la presentación de un proyecto (título, fundamentación, objetivos, etc)</p> <p>Criterios de evaluación. Determinación de los parámetros a evaluar. Parámetros típicos de acuerdo a la clase de elementos a evaluar.</p> <p>Homogeneización de parámetros. Ponderación de parámetros. Matriz de evaluación. Alternativas, conclusiones e informes. Evaluaciones integrales.</p> <p>Integración de hardware, firmware y software. Distintos grados de integración. Evaluación de los mismos. Aplicación de la metodología top down a la evaluación. Técnicas asociadas: Benchmarks. Evaluación de hardware. Determinación de los objetivos. Distintos tipo de evaluación según las prestaciones y según los requerimientos. Parámetros típicos en cada uno de los casos. Influencia del firmware para un hardware determinado.</p> <p>Aplicación a los productos existentes en el mercado. Evaluación de hardware de redes. Determinación de la topología. Distintos tipo de requerimientos. Componentes típicos de cada tipo red. Parámetros típicos en cada uno de los casos. Influencia del volumen de transacciones.</p> <p>Aplicación a los productos existentes en el mercado. Evaluación de software de base. Determinación de los objetivos. Distintos métodos de</p>	

evaluación según las funciones y según los requerimientos. Parámetros típicos en cada caso. Aplicación a los sistemas de base existentes en el mercado.

Evaluación de software de aplicación. Concepto de producto de software. Estudio de factibilidad para productos de software. Análisis comparativo entre productos similares. Matrices comparativas. Análisis de aplicabilidad y transportabilidad de distintos productos disponibles en el mercado. Evaluación.

Área de organización y sistemas. Inserción en la estructura de la Organización. Estructura del área de Organización y Sistemas. Departamentalización. Objetivos y funciones de cada departamento. Adaptación de la estructura según los requerimientos. Organización por tarea. Organización por proyecto. Misiones y Funciones. Capacitación necesaria de los responsables de cada puesto. Auditoría de sistemas. Controles en el procesamiento de datos. La auditoría como función de control. Control en el desarrollo de sistemas y en los programas. Control de entrada y salida de datos y documentos. Control operativo y funcional. Utilización de la computadora en la auditoría. El equipo de auditoría. Ubicación jerárquica y estructura funcional. Función del auditor en los sistemas de información. Técnicas para la auditoría de sistemas.

ORIENTACIONES DIDÁCTICAS

- Elaboración de un proyecto
- Seguimiento y evaluación del mismo.
- Representar gráficamente el organigrama de una empresa donde realizan las prácticas profesionalizantes.

<p>7° AÑO</p> <p>FORMACIÓN CIENTÍFICO TECNOLÓGICA</p> <p>MODELOS Y SISTEMAS II</p>	
CARGA HORARIA	<p>3 Horas Semanales</p> <p>108 Horas Anuales</p>
CAPACIDADES A DESARROLLAR	<p>Finalizado el cursado de la materia los y las estudiantes deberán haber desarrollado las siguientes Capacidades:</p> <ul style="list-style-type: none"> - Resolver situaciones problemáticas en el marco de las diferentes aplicaciones. - Interpretar consignas. - Resolver algoritmos en los que se apliquen estructuras condicionales. - Identificar métodos, para su correcta aplicación en diferentes situaciones (tomadas de la realidad). - Resolver problemas simples mediante diagramación lógica. - Seleccionar la herramienta adecuada según la plataforma sobre la cual se realizará la solución. - Diseñar y resolver la documentación de proyectos. - Diseñar manuales y utilizar técnicas pertinentes de los modelos de representación de sistemas. - Elaborar documentación pertinente. - Implementar y evaluar la solución desarrollada
CONTENIDOS	
<p>Herramientas de Diagramas: de flujo de datos, de control. Diccionario de datos. Lógica de los procesos. Almacenamiento de datos. Convenciones. Diagrama de transición de estado. Diagrama de entidad. Relación. Técnicas para el Análisis: Desarrollo control y seguimiento del proyecto. Prototipo. Ciclo de vida. Métodos de asignación y transporte: Método húngaro. Aproximación por costos mínimos. Soluciones alternativas para cada caso.</p>	

Otros métodos. Números al Azar: Funciones generadoras de números aleatorios. Métodos, manuales y algoritmos. Comparación. Teoría de los modelos: Tabla y árboles de decisión. Abstracción de datos. Objetivos en el uso de modelos matemáticos. Clasificación de modelos. Predicción de los resultados. Modelos de experimentación controlados. Diseño de documentación y manuales de sistemas y usuarios: Técnicas para la elaboración de la documentación de las tareas. Lineamientos generales para la confección del manual. Conformación de la organización de la documentación. Grafos y redes: Problemas básicos. Transportes. Flujo máximo. Ruta más corta. Camino mínimo. Camino crítico. Algoritmos básicos

ORIENTACIONES DIDÁCTICAS

- Generar consignas que incluyan conocimientos previos de otras materias, que puedan utilizarse como base de aplicación en consignas propias de esta materia.
- Poner en evidencia la relación teoría y práctica, con la utilización de anotaciones en carpeta, para consulta de las resoluciones.
- Leer y analizar situaciones problemáticas, para luego resolver paso a paso.
- Utilizar material multimedia como recurso didáctico para presentación de temática en curso.
- Utilizar registros como parte del análisis de las consignas, para una correcta interpretación.

<div> <div>7° AÑO</div> <div>FORMACIÓN CIENTÍFICO TECNOLÓGICA</div> </div> <div>ORGANIZACIÓN Y MÉTODOS</div>	
CARGA HORARIA	<div>2 Horas Semanales</div> <div>72 Horas Anuales</div>
<div> <div>CAPACIDADES A DESARROLLAR</div> </div>	<p>Finalizado el cursado de la materia los y las estudiantes deberán haber desarrollado las siguientes Capacidades:</p> <ul style="list-style-type: none"> - Identificar aspectos críticos. - Dividir la asignación en subtarear o productos intermedios. - Establecer un orden o secuencia de trabajo. - Estimar tiempos de realización. - Utilizar metodologías de búsqueda de información de fuentes confiables. - Aplicar técnicas y metodologías para la resolución de problemas.
	<div>CONTENIDOS</div> <p>Conocimiento de la labor industrial y de los sistemas productivos. Definición de las funciones del área Industrial. Departamentos. Tecnologías de fabricación y sistemas productivos de tecnologías específicas de software. Ciclos de trabajo. Estudio de métodos, tiempos y movimientos. Normalización. Distribución de planta. Diagramas de áreas y equipos. Higiene y seguridad: conceptos básicos. Atención técnica al cliente. Aplicación de los conocimientos adquiridos a las empresas de servicios_ Producción de textos. Manuales del usuario, manual del programador.</p>

ORIENTACIONES DIDÁCTICAS

- Resolución de ejercicios y problemas
- Adquisición de aprendizajes mediante el estudio de casos reales o simulados
- Realización de un proyecto para la resolución de un problema aplicando habilidades y conocimientos adquiridos
- Debate, diálogo, exposición, demostración

<div> <div>7° AÑO</div> <div>FORMACIÓN TÉCNICA ESPECÍFICA</div> </div> <div> <div>PROYECTO INTEGRADOR</div> </div>	
<div>CARGA HORARIA</div>	<div> <div>4 Horas Semanales</div> <div>144 Horas Anuales</div> </div>
<div> <div>CAPACIDADES A DESARROLLAR</div> </div>	<div> <div>Finalizado el cursado de la materia los y las estudiantes deberán haber desarrollado las siguientes Capacidades:</div> <ul style="list-style-type: none"> - Conocer las herramientas, lenguajes, librerías y frameworks para el desarrollo de aplicaciones web. - Seleccionar la mejor opción de acuerdo a las dimensiones del proyecto a desarrollar. - Desarrollar aplicaciones web bajo el patrón MVC e integración con clientes Javascript y diferentes bases de datos. - Detectar fallas en el sistema a través de herramientas de testing y corrección de las mismas. </div>
<div>CONTENIDOS</div>	<div> <div>Programación orientada a objetos. Objetos, métodos y eventos. Clases, herencia y polimorfismo. Herramientas y entornos de trabajo del lenguaje seleccionado. Desarrollo de aplicaciones por capas. Desarrollo de interfaces visuales. Ajax, Json y XML.</div> <div>Frameworks en cliente (jQueryUI, JQueryEasyUI, Angular, React, etc), controles Javascript e interface con servidor. Templates en servidor para la separación de código del HTML, CSS y Javascript. Utilización del lenguaje SQL, conexiones a bases de datos, listado de datos en el cliente, carga y edición de datos por formularios, búsquedas y reportes web en HTML o pdf utilizando desarrollos en dos capas. Patrón Modelo, Vista, Controlador. Herramientas para instalación de software por dependencia (por ejemplo composer en PHP). Frameworks de trabajo en el servidor para el lenguaje seleccionado. Implementación de un framework actual para el desarrollo en tres capas con el patrón MVC. Herramientas de testing y documentación. Capacitación y asistencia al usuario. Detección y corrección de errores</div> </div>

ORIENTACIONES DIDÁCTICAS

- Proyectos interdisciplinarios con trabajos especiales en las diferentes materias de 7 año.
- Armado de propuestas digitales y gráficas en función de desarrollos completos en programación.
- Inclusión de trabajos articulando con Practicas Profesionalizantes.

7° AÑO FORMACIÓN TÉCNICA ESPECÍFICA	
DESARROLLO DE SOFTWARE PARA PLATAFORMAS MÓVILES	
CARGA HORARIA	4 Horas Semanales 144 Horas Anuales
CAPACIDADES A DESARROLLAR	<p>Finalizado el cursado de la materia los y las estudiantes deberán haber desarrollado las siguientes Capacidades:</p> <ul style="list-style-type: none"> - Plantear un diseño de plataforma móvil de acuerdo al tema a presentar. - Reconocer y utilizar el lenguaje de estilos de presentación (css) - JavaScript, PHP y MySQL, - Comprender los algoritmos que, aplicados a Processing, producirán gráficos y sonidos. - Identificar web 2.0 y dinámica. - Analizar y resolver diferentes situaciones problemáticas.
CONTENIDOS	
<p>Nociones básicas de los lenguajes de programación involucrados en la creación de páginas web dinámicas: JavaScript, PHP y MySQL, abriendo así la posibilidad de crear aplicaciones creativas en línea. Introduce también la programación para plataformas móviles. Se aplicará la programación a la creación de formas audiovisuales generativas. Se estudiarán diferentes algoritmos que, aplicados a Processing, producirán gráficos y sonidos producidos como resultado de su ejecución. Se introducirán una serie de herramientas que se pueden instalar y configurar fácil y rápidamente para crear aplicaciones web dinámicas de todo tipo: blogs, wikis y más. Introducción a la llamada Web 2.0, o Web dinámica, en la que se analizarán servicios como YouTube, Facebook, Flickr, y otros, haciendo prácticas de desarrollo sobre sus interfaces de programación para crear nuevas aplicaciones, comunmente llamadas "mashups"</p>	

ORIENTACIONES DIDÁCTICAS

- Desarrollo de plataformas móviles para diferentes proyectos.
- Trabajos conjuntos con prácticas profesionalizantes para ser utilizados según requerimientos de la misma
- Se presentarán situaciones problemáticas que faciliten el trabajo colaborativo y el debate sobre estrategias resolutivas.
- Diseñar y crear algoritmos de manera autónoma.
- Aplicar los contenidos en contextos reales.

7° AÑO FORMACIÓN TÉCNICA ESPECÍFICA	
DISEÑO E IMPLEMENTACIÓN DE SITIOS WEB	
CARGA HORARIA	4 Horas Semanales 144 Horas Anuales
CAPACIDADES A DESARROLLAR	Finalizado el cursado de la materia los y las estudiantes deberán haber desarrollado las siguientes Capacidades: <ul style="list-style-type: none"> - Fortalecer los contenidos para el desarrollo web. - Propiciar el desarrollo de las competencias necesarias que permitan al alumno evaluar servicios para encontrar una solución óptima de acuerdo con las demandas de sistemas. - Estimular la cultura participativa entre docentes y alumnos
CONTENIDOS	
Seguridad – Página Web: Seguridad de Sitio web: Https. Comercio digital. Token. Vulnerabilidades y otros errores. Desarrollo de páginas Web. Marketing Digital. SEO. Análisis de Métricas de Google Analytics. Gestores de Contenidos: WordPress, Wix, Joomla, entre otros. Aplicaciones Dinámicas. Estudio de desarrollo de servicios como Youtube, Facebook, Flickr entre otros. Resolución por computadora. Aplicaciones dinámicas 2. Integrar lenguajes: Php, Java, Css, HTML, MySQL. Determinar y Documentar las fases del desarrollo. Resolución por computadora. Actualizar página web. Algorítmica Audiovisual Arte generativo: antecedentes y perspectivas. Resolución por computadora: Lenguajes de programación: Processing	

ORIENTACIONES DIDÁCTICAS

- En cada contenido teórico representar un caso de uso real aplicado a empresas o instrumentos de uso diario.
- Formulación e implementación de soluciones informáticas

7° AÑO PRÁCTICAS PROFESIONALIZANTES		PRACTICAS PROFESIONALIZANTES DEL SECTOR PROGRAMACIÓN	
CARGA HORARIA		216 Horas Anuales	
CAPACIDADES A DESARROLLAR		<p>Las Prácticas Profesionalizantes son aquellas estrategias formativas integradas en la propuesta curricular, con el propósito de que los estudiantes consoliden, integren y amplíen, las capacidades y saberes que se corresponden con el perfil profesional en el que se están formando, organizadas por la institución educativa, referenciadas en situaciones de trabajo y/o desarrolladas dentro o fuera de la escuela.</p> <p>Su propósito es poner en práctica saberes profesionales significativos sobre procesos socio-productivos de bienes y servicios, que tengan afinidad con el futuro entorno de trabajo en cuanto a su sustento científico, tecnológico y técnico.</p>	
CONTENIDOS		<p>Al diseñar las Prácticas Profesionalizantes, las instituciones tendrán como intención:</p> <ul style="list-style-type: none"> - Fortalecer los procesos educativos a través de instancias de encuentro y realimentación mutua con organismos del sector socio productivo y/o entidades de la comunidad. - Fomentar la apertura y participación de la institución en la comunidad. - Establecer puentes que faciliten a los estudiantes la transición desde la escuela al mundo del trabajo y a los estudios superiores. - Impulsar el reconocimiento de las demandas del contexto productivo local. 	

Objetivos

A través de las Prácticas Profesionalizantes los alumnos tendrán oportunidades de:

- Reflexionar críticamente sobre su futura práctica profesional, sus resultados objetivos e impactos sobre la realidad social.
- Enfrentarse a situaciones de incertidumbre, singularidad y conflicto de valores.
- Integrar y transferir aprendizajes adquiridos a lo largo del proceso de formación.
- Reconocer y valorar el trabajo decente en el marco de los Derechos de los Trabajadores y del respeto por las condiciones de higiene y seguridad en que debe desarrollarse.
- Formar integralmente a un ciudadano para ejercer responsablemente sus deberes y derechos, complementando a su profesionalidad específica.

ORIENTACIONES DIDÁCTICAS

Las Prácticas Profesionalizantes pretenden familiarizar e introducir a los estudiantes en los procesos y el ejercicio profesional vigentes, para lo cual utilizan un variado tipo de estrategias didácticas ligadas a la dinámica profesional caracterizada por la incertidumbre, la singularidad y el conflicto de valores.

En el marco de la Educación Técnico Profesional, estas prácticas formativas deben ser concebidas como el núcleo central y al mismo tiempo, como eje transversal de la formación, que da sentido al conjunto saberes y capacidades que comprenden un título técnico.

Organización y Contexto

Las Prácticas Profesionalizantes abren un abanico de posibilidades para realizar experiencias formativas en distintos contextos y entornos de aprendizaje.

En relación con el contexto de implementación, las prácticas se pueden desarrollar:

- Dentro de la institución educativa.
- Fuera de la institución educativa.

En relación con el entorno de implementación, las prácticas se pueden desarrollar:

- En el entorno de la institución escolar (Proyectos de Prácticas Profesionalizantes, Proyectos Tecnológicos, Módulos Integradores del 7mo. Año, Trabajos por Cuenta de Terceros, entre otros).

ANEXO

ENTORNOS FORMATIVOS

Se centran en identificar la infraestructura, el equipamiento y las instalaciones a los cuales los alumnos deberían tener acceso para desarrollar las capacidades necesarias en su trayectoria formativa.

Es necesario que toda infraestructura y equipamiento debe tener una clara correspondencia con el proyecto educativo, las situaciones de enseñanza y el desarrollo de las actividades que los alumnos realizan, atendiendo a los propósitos del Nivel Secundario.

El equipamiento, los insumos, dispositivos y demás componentes básicos en los entornos formativos serán establecidos por la Dirección de Educación Técnica y pueden formar parte del proceso de homologación de títulos.

En todos los casos, los ambientes deben cumplir con las condiciones de higiene y seguridad.

Son entornos formativos para el ciclo superior de educación técnica de la presente tecnicatura:

1. Laboratorio de Hardware
2. Laboratorio de Redes
3. Laboratorio de Programación
4. Laboratorio de Software



G O B I E R N O D E L A P R O V I N C I A D E B U E N O S A I R E S

Hoja Adicional de Firmas
Anexo

Número: IF-2018-31697559-GDEBA-DETECDGCYE

LA PLATA, BUENOS AIRES
Viernes 14 de Diciembre de 2018

Referencia: DISEÑO PROGRAMACION

El documento fue importado por el sistema GEDO con un total de 93 pagina/s.

Digitally signed by CABALLIN Cesar Ceferino
Date: 2018.12.14 16:01:17 ART
Location: Provincia de Buenos Aires

Cesar Caballin
Director
Dirección de Educación Técnica (Docente)
Dirección General de Cultura y Educación

Digitally signed by GDE BUENOS AIRES
DN: cn=GDE BUENOS AIRES, c=AR, o=MINISTERIO DE
JEFATURA DE GABINETE DE MINISTROS BS AS,
ou=SUBSECRETARIA para la MODERNIZACION DEL
ESTADO, serialNumber=CUIT 30715471511
Date: 2018.12.14 16:01:12 -03'00'