

**DIRECCIÓN DE FORMACIÓN PROFESIONAL
PROGRAMA DE FORMACIÓN TÉCNICO PROFESIONAL**

Certificado de Aptitud profesional en

Programación básica de computadora

Estrategia de formación: Habilitación

Santo Domingo, R. D.

16/09/2024

Diseño adaptado al Marco Nacional de Cualificaciones de la República Dominicana



Índice

1.	PRESENTACIÓN:	3
2.	DATOS GENERALES DE LA CUALIFICACIÓN: CERTIFICADO DE APTITUD PROFESIONAL EN PROGRAMACIÓN BÁSICA DE COMPUTADORA:	5
2.1	DENOMINACIÓN:	5
2.2	NIVEL DE LA CUALIFICACIÓN:	5
2.3	FAMILIA PROFESIONAL:	5
2.4	CÓDIGO MNC-RD:	5
2.5	DURACIÓN TOTAL	5
2.6	HORAS TRANSVERSALES GENÉRICAS	5
2.7	HORA FORMAC. ASOC. AL PERFIL PROFESIONAL	5
2.8	CÓDIGO INFOTEP:	5
2.9	CÓDIGO CNEF-RD 2019:	5
2.10	FECHA DE APROBACIÓN:	5
2.11	TIEMPO DE REVISIÓN:	5
2.12	NO. DE REVISIÓN:	5
3.	PERFIL PROFESIONAL	6
3.1	COMPETENCIA GENERAL DE LA CUALIFICACIÓN:	6
3.2	UNIDADES DE COMPETENCIA:	6
	ELEMENTOS DE COMPETENCIA	6
	CRITERIOS DE DESEMPEÑO	6
	RESULTADOS DE APRENDIZAJE	6
	CRITERIOS DE EVALUACIÓN	6
3.3	ENTORNO PROFESIONAL:	22
4	FORMACIÓN ASOCIADA AL PERFIL	23
5	DESCRIPCIÓN DE LOS MÓDULOS DE APRENDIZAJE	25
	MATEMÁTICA DISCRETA	25
	INTRODUCCIÓN A LA COMPUTADORA Y SISTEMA OPERATIVO	31
	INTRODUCCIÓN A LA PROGRAMACIÓN	39
	ANÁLISIS Y DISEÑO DE SISTEMA I	43
	INTRODUCCIÓN A LA BASE DE DATOS	51
	PROGRAMACIÓN EN LENGUAJE C++	56
	PROGRAMACIÓN C#	64
6	PERFIL TÉCNICO PEDAGÓGICO DEL FACILITADOR (A) DEL PROGRAMA	69
7	REQUERIMIENTO DE RECURSOS	70
8	LISTA MAESTRA	71
9	REFERENCIAS DOCUMENTALES	72
10	RESPONSABLES DEL DISEÑO	73
11	SUGERENCIAS DE MEJORA PARA LA REVISIÓN DEL PROGRAMA DE FORMACIÓN	74
12	GUÍA DE PRÁCTICAS POR MÓDULOS	76

1. PRESENTACIÓN:

El presente Programa de Formación, **Programación básica de computadora, Certificado de Aptitud profesional**, ha sido revisado por competencia centrado en resultados de aprendizaje. Tiene como objetivo lograr en los participantes las competencias requeridas en el campo laboral, a través de la adquisición de conocimientos, habilidades conductuales y Responsabilidad y autonomía. El mismo es un instrumento de apoyo a la labor docente, orienta la conducción del proceso enseñanza y aprendizaje y contribuye a mejorar la calidad y la pertinencia de la formación.

El programa contiene los siguientes apartados:

DATOS GENERALES DE LA CUALIFICACIÓN, que incluye: Denominación, Nivel de cualificación, Familia profesional, Código MNC-RD, Duración total, Código INFOTEP, Código CNEF-2019, Fecha de aprobación, Tiempo de revisión Población enfocada, Requisitos de entrada, Estrategia de ejecución y Tipo de cualificación.

PERFIL PROFESIONAL, que se compone de la Competencia general de la cualificación de competencias, las Unidades de competencias transversales, Unidades de competencias genéricas o básicas, Unidades de competencias específicas y técnicas, Elementos de competencia, Criterios de desempeño, Resultados de aprendizaje, Criterios de evaluación y el Entorno profesional.

FORMACIÓN ASOCIADA AL PERFIL, que incluye los módulos que componen el programa de la cualificación.

MÓDULOS DE APRENDIZAJE, que se componen de: Código del módulo, Denominación del módulo, Correspondencia con la Unidad de Competencia, Competencia final del Módulo, Duración en horas, Resultados de Aprendizaje, Contenidos: Saber (conocimientos), Saber hacer (Habilidades cognitivas y prácticas), Saber ser (Habilidades conductuales), Criterios de evaluación y Estrategias metodológicas para el planeamiento didáctico.

REQUERIMIENTO DE RECURSOS (ambiente de formación).

LISTA MAESTRA, se enumeran todos los Materiales, Equipos, Herramientas, Mobiliarios e Insumos que serán utilizados en el desarrollo de la cualificación.

REFERENCIAS DOCUMENTALES, que incluye: las referencias bibliográficas, hemerográficas o de cualquier tipo de publicación, espacios electrónicos y software que sirven de apoyo al docente y al participante.

Para la aplicación del presente programa de formación se recomienda el desarrollo de las siguientes estrategias generales de enseñanza, aprendizaje y evaluación.

- La identificación de conocimientos, habilidades y destrezas que domina el participante ya sea por formación o experiencia antes de iniciar el curso de capacitación, como una forma de estructurar su plan de formación personalizado.
- Propiciar el desarrollo de la autonomía en el proceso de enseñanza y aprendizaje, mediante técnicas y estrategias que fomenten el aprendizaje activo y centrado en los participantes y de esta forma lograr el fortalecimiento de habilidades blandas tales como la toma de decisiones, la resolución de problemas, el pensamiento crítico, trabajo en equipo, la creatividad, la investigación, entre otras.

- Ofrecer distintas actividades de enseñanza y aprendizaje donde el participante tenga la oportunidad de elegir, aquel dónde desarrollar sus competencias sea más familiar y responda a sus necesidades de formación y tipo de aprendizaje.
- Fomentar la construcción o refuerzo de conocimientos, habilidades, destrezas y actitudes, con la investigación documental y de campo, el trabajo en equipo, visitas al sector productivo o ámbitos relacionados, aprovechando estas experiencias, para incorporar nuevos aprendizajes a los ya existentes en el participante.
- Fomentar el desarrollo y dominio de habilidades y destrezas con ejercicios y prácticas frecuentes, realización de proyectos, en los que se crean condiciones reales o simuladas para la resolución de problemas, analizar casos, o repetir rutinas de trabajo en donde el participante tenga la oportunidad de transferir los aprendizajes logrados a contextos diferentes.
- Incorporar en la evaluación de los aprendizajes el uso de instrumentos de evaluación tales como la rúbrica, proyectos, portafolios físicos y digitales.

Este programa debe someterse a revisión cada tres (3) años, la Dirección de Formación Profesional, a través del Departamento de Desarrollo Curricular le dará seguimiento a su aplicación a fin de hacer los ajustes y las enmiendas que requiera, acorde con los avances tecnológicos del área. Con tal propósito, agradecemos las observaciones que nos remitan en el formulario anexo. El mismo podrá ser completado, según se desarrolle el curso y luego hacerlo llegar a la Dirección de Formación Profesional.

2. DATOS GENERALES DE LA CUALIFICACIÓN: Certificado de Aptitud profesional en Programación básica de computadora.

2.1 DENOMINACIÓN:	Programación básica de computadora				
2.2 NIVEL DE LA CUALIFICACIÓN:	N/A	2.3 FAMILIA PROFESIONAL:	INCO (Informática y comunicaciones)		
2.4 CÓDIGO MNC-RD:	N/A				
2.5 DURACIÓN TOTAL	370 horas	2.6 HORAS TRANSVERSALES GENÉRICAS	35 horas	2.7 HORA FORMAC. ASOC. AL PERFIL PROFESIONAL	335 horas
2.8 CÓDIGO INFOTEP:	30936	2.9 CÓDIGO CNEF-RD 2019:		0612 (Diseño y administración de redes y bases de datos)	
2.10 FECHA DE APROBACIÓN:	16/09/2024	2.11 TIEMPO DE REVISIÓN:		3 años	2.12 NO. DE REVISIÓN: 3
2.13 POBLACIÓN ENFOCADA: Personas jóvenes y adultas de ambos sexos, con actitudes e interés en desarrollarse en la cualificación de Programación básica de computadora.					
2.14 REQUISITOS DE INGRESO: 1. Certificado de haber superado el 1er. Ciclo de secundaria o (estar cursando de 2do. Hasta 4to. De bachiller antiguo sistema).. 2. Tener 16 años o más.					
2.15 ESTRATEGIA DE FORMACIÓN:	Habilitación		2.16 TIPO DE CUALIFICACIÓN:		Nacional

Nota1: Los módulos transversales genéricos tipos MOOC (Cursos Masivos Abiertos en Línea) que se ejecutaran a través de la plataforma de Infotep Virtual, **son módulos obligatorios para obtener el título de Técnico**, estos podrán desarrollarse de manera simultánea o antes de iniciar con los módulos técnicos. Por tal motivo, a la hora de planificar el programa de formación, se deben contemplar para su ejecución.

- Formación humana.
- Protección al medio ambiente y cambio climático.

3. PERFIL PROFESIONAL

3.1 COMPETENCIA GENERAL DE LA CUALIFICACIÓN:

Al finalizar la cualificación la persona participante estará en la capacidad de manejar los fundamentos básicos de programación para aplicarlo en su entorno visual, en un ambiente de confianza y participación.

3.2 UNIDADES DE COMPETENCIA:

El o la participante será competente si domina las siguientes unidades de competencias:

Unidades de competencias básicas	Elementos de competencia	Criterios de desempeño	Resultados de aprendizaje	Criterios de evaluación
1. Desarrollar diferentes operaciones algebraicas a través de análisis y posibilidades lógicas.	-	-	1. Realiza operaciones entre conjuntos, para demostrar diversos teoremas relacionados con los números enteros.	La persona participante: - Identifica los diferentes tipos de números (naturales, enteros, fraccionarios y decimales), según asignaciones. - Realiza conversiones entre fracciones y números decimales, según procedimientos. - Simplifica y reduce fracciones con distintos denominadores, según indicaciones. - Evidencia responsabilidad mientras realiza operaciones entre conjuntos, según procedimientos requeridos.
	-	-	2. Realiza operaciones de potenciación, para calcular potencia de base y exponente natural.	La persona participante: - Identificar los diferentes tipos de dispositivos eléctricos, según procedimientos. - Realiza operaciones de factores iguales, según lo requerido. - Calcula potencia de base entera y exponente natural, según procedimientos exigidos. - Realiza operaciones de números enteros y racionales con radicales, conforme a requerimientos. - Evidencia participación mientras realiza operaciones de potenciación, según procedimientos establecidos.

Unidades de competencias básicas	Elementos de competencia	Criterios de desempeño	Resultados de aprendizaje	Criterios de evaluación
	-	-	3. Organiza ecuaciones algebraicas, para aplicar operaciones fundamentales en el análisis y solución de problemas matemáticos.	La persona participante: - Analiza sobre la clasificación y organización de ecuaciones algebraicas, según asignaciones. - Clasifica ecuaciones algebraicas, según lo requerido. - Realiza ecuaciones algebraicas, según requerimientos. - Resuelve ecuaciones de primer grado con una incógnita, conforme a lo requerido. - Realiza despeje de fórmulas algebraicas, según proceso y requerimientos. - Evidencia responsabilidad y disciplina mientras organiza ecuaciones algebraicas, según procedimientos requeridos.

Nota: Los módulos básicos y transversales, no se le incluyen los elementos de competencia ni los criterios de desempeño.

Unidades de competencias técnicas	Elementos de competencia	Criterios de desempeño	Resultados de aprendizaje	Criterios de evaluación
1. Instalar sistemas operativos realizando tareas comunes.	1.1. Configurar y mantener los componentes de hardware de un sistema de computación, según exigencias requeridas.	- Configurar las motherboards, procesadores y módulos de memoria (RAM y ROM), según especificaciones técnicas. - Utilizar herramientas de diagnóstico para identificar fallas en los sistemas de enfriamiento y las tarjetas adaptadoras, siguiendo especificaciones técnicas. - Desarrollar y ejecutar un plan de mantenimiento preventivo periódico que incluye la revisión de todos los componentes de hardware, según procedimientos establecidos.	1. Configura y mantiene los componentes de hardware de un sistema de computación, para diagnosticar fallas y realizar el mantenimiento preventivo y correctivo.	La persona participante: - Define concepto de sistema de computación, según explicaciones dadas. - Configura las motherboards, procesadores y módulos de memoria (RAM y ROM), según especificaciones técnicas. - Utiliza herramientas de diagnóstico para identificar fallas en los sistemas de enfriamiento y las tarjetas adaptadoras, siguiendo especificaciones técnicas. - Desarrolla y ejecuta un plan de mantenimiento preventivo periódico que incluye la revisión de todos los componentes de hardware, según procedimientos establecidos. - Evidencia responsabilidad y adhesión a las normas mientras configura y mantiene los componentes de hardware de un sistema de computación, según exigencias requeridas.
	1.2. Instalar sistema operativo, según estándares de informática.	- Realizar la instalación personalizada de un sistema operativo, seleccionando adecuadamente las particiones del disco y configurando opciones avanzadas, según los requisitos del cliente. - Configurar los archivos de secuencia de inicio y archivos de registro del sistema operativo para optimizar el tiempo de arranque, según especificaciones técnicas. - Desarrollar y aplicar procedimientos de soporte técnico post-instalación para asegurar la correcta operación del sistema operativo, según especificaciones técnicas.	2. Instala sistemas operativos, para manejar procedimientos de selección, instalación, configuración y soporte.	La persona participante: - Identifica las características de los sistemas operativos modernos, incluyendo los de red y de escritorio, según requerimientos. - Realiza la instalación personalizada de un sistema operativo, seleccionando adecuadamente las particiones del disco y configurando opciones avanzadas según los requisitos del cliente. - Configura los archivos de secuencia de inicio y archivos de registro del sistema operativo para optimizar el tiempo de arranque, según especificaciones técnicas. - Desarrolla y aplica procedimientos de soporte técnico post-instalación para asegurar la correcta operación del sistema operativo, según especificaciones técnicas. - Evidencia responsabilidad mientras instala sistema operativo, según estándares de informática.

Unidades de competencias técnicas	Elementos de competencia	Criterios de desempeño	Resultados de aprendizaje	Criterios de evaluación
	1.3. Configurar y optimizar sistemas operativos Windows, Linux y Mac OS, según los estándares de calidad y procedimientos establecidos.	- Ejecutar actualizaciones del sistema operativo sin comprometer la integridad del sistema, según los procedimientos de seguridad establecidos. - Realizar la resolución de problemas de los sistemas operativos, según los protocolos de diagnóstico y reparación definido. - Realizar la instalación de dispositivos de hardware en sistemas operativos Windows, Linux y Mac OS, según lo requerido. - Desarrollar y ejecutar un plan de optimización del sistema operativo, utilizando herramientas del sistema y software de terceros para mejorar el rendimiento general del equipo, de acuerdo a especificaciones técnicas.	3. Configura y optimiza sistemas operativos Windows, Linux y Mac OS, para asegurar el correcto funcionamiento de los equipos y maximizar el rendimiento y la seguridad del sistema.	La persona participante: - Identifica correctamente las interfaces de Windows, Linux y Mac OS, de acuerdo a explicaciones. - Ejecuta actualizaciones del sistema operativo sin comprometer la integridad del sistema, según los procedimientos de seguridad establecidos. - Realiza la resolución de problemas de los sistemas operativos, según los protocolos de diagnóstico y reparación definido. - Realiza la instalación de dispositivos de hardware en sistemas operativos Windows, Linux y Mac OS, según lo requerido. - Desarrolla y ejecuta un plan de optimización del sistema operativo, utilizando herramientas del sistema y software de terceros para mejorar el rendimiento general del equipo, de acuerdo a especificaciones técnicas. - Evidencia orientación a la calidad y al detalle mientras configura y optimiza sistemas operativos Windows, Linux y Mac OS, según los estándares de calidad y procedimientos establecidos.
	1.4. Operar sistemas operativos y herramientas de gestión de archivos, según las políticas de seguridad.	- Realizar copias de seguridad y restauración de datos en Windows y Linux, según procedimientos establecidos. - Configurar y utilizar las utilidades de administración del sistema en Windows, Unix, Gnu/Linux y Mac OS, según las especificaciones de seguridad. - Crear, eliminar y organizar archivos y carpetas en el Explorador de Windows y en la línea de comandos (cmd) de Windows, según los protocolos de gestión de archivos.	4. Opera sistemas operativos y herramientas de gestión de archivos, para administrar y mantener el equipo de manera eficiente.	La persona participante: - Identifica los elementos del explorador de Windows, de acuerdo a instrucciones. - Realiza copias de seguridad y restauración de datos en Windows y Linux, según procedimientos establecidos. - Configura y utiliza las utilidades de administración del sistema en Windows, Unix, Gnu/Linux y Mac OS, según las especificaciones de seguridad. - Crea, elimina, y organiza archivos y carpetas en el Explorador de Windows y en la línea de comandos (cmd) de Windows, según los protocolos de gestión de archivos. - Evidencia responsabilidad y adhesión a las normas mientras opera sistemas operativos y herramientas de gestión de archivos, según las políticas de seguridad.

Unidades de competencias técnicas	Elementos de competencia	Criterios de desempeño	Resultados de aprendizaje	Criterios de evaluación
2. Manejar lógica de programación, utilizando algoritmos y elaborando diagrama de flujo	2.1. Desarrollar la lógica de programación utilizando los ciclos básicos del procesamiento de información, según los estándares de programación.	- Seleccionar y utilizar adecuadamente tipos de variables y constantes, según las necesidades de almacenamiento y procesamiento de datos. - Diseñar estructuras de control condicionales y ciclos repetitivos, según los objetivos del programa. - Codificar estructuras de control condicionales y ciclos repetitivos, según los objetivos del programa. - Utilizar diagramas UML para modelar requerimientos de negocio y estructuras de control, según las especificaciones del cliente.	1. Desarrolla la lógica de programación utilizando los ciclos básicos del procesamiento de información, para aplicar principios de programación estructurada y el manejo eficiente de datos.	La persona participante: - Explica la evolución histórica de la programación y su impacto en el desarrollo de software, según los principios básicos de procesamiento de información. - Selecciona y utiliza adecuadamente tipos de variables y constantes, según las necesidades de almacenamiento y procesamiento de datos. - Diseña y codifica estructuras de control condicionales y ciclos repetitivos, según los objetivos del programa. - Utiliza diagramas UML para modelar requerimientos de negocio y estructuras de control, según las especificaciones del cliente. - Evidencia pensamiento lógico y orientación a la calidad mientras desarrolla la lógica de programación utilizando los ciclos básicos del procesamiento de información, según los estándares de programación.
	2.2. Desarrollar diagramas de flujo y algoritmos, según procedimientos.	- Desarrollar algoritmos que utilicen vectores y matrices para almacenar y manipular datos, según las características del problema. - Optimizar algoritmos mediante el uso adecuado de estructuras de control y ciclos anidados, según los criterios de eficiencia y rendimiento del programa. - Corregir errores en los algoritmos y diagramas de flujo desarrollados, aplicando técnicas de depuración y pruebas, según los estándares de calidad del software	2. Desarrolla diagramas de flujo y algoritmos, para implementar soluciones eficientes en procesos informáticos.	La persona participante: - Analiza sobre elaboración de diagrama de flujo y algoritmos, según procedimientos. - Desarrolla algoritmos que utilicen vectores y matrices para almacenar y manipular datos, según las características del problema. - Optimiza algoritmos mediante el uso adecuado de estructuras de control y ciclos anidados, según los criterios de eficiencia y rendimiento del programa. - Corrige errores en los algoritmos y diagramas de flujo desarrollados, aplicando técnicas de depuración y pruebas, según los estándares de calidad del software - Evidencia pensamiento lógico y creatividad mientras desarrolla diagramas de flujo y algoritmos, según procedimientos.

Unidades de competencias técnicas	Elementos de competencia	Criterios de desempeño	Resultados de aprendizaje	Criterios de evaluación
	2.3. Desarrollar el diseño de un sistema de base de datos, según los principios de modelado de datos y los requerimientos del negocio.	- Recopilar información con los usuarios del negocio para especificar los requisitos de la base de datos, utilizando técnicas de especificación de requisitos adecuadas, según las necesidades del negocio. - Diseñar el modelo conceptual de una base de datos utilizando el enfoque entidad-relación (ER), identificando entidades, atributos y relaciones, según los requisitos del negocio. - Evaluar el sistema de base de datos durante su ciclo de vida, incluyendo la revisión periódica del diseño y la aplicación de mejoras basadas en el feedback de los usuarios y el rendimiento del sistema, según los estándares de calidad de la empresa. - Ajustar el sistema de base de datos durante su ciclo de vida, incluyendo la revisión periódica del diseño y la aplicación de mejoras basadas en el feedback de los usuarios y el rendimiento del sistema, según los estándares de calidad de la empresa.	3. Desarrolla el diseño de un sistema de base de datos, para realizar su implementación y ajuste.	La persona participante: - Define concepto de datos y de bases de datos, según procedimientos. - Recopila información con los usuarios del negocio para especificar los requisitos de la base de datos, utilizando técnicas de especificación de requisitos adecuadas, según las necesidades del negocio. - Diseña el modelo conceptual de una base de datos utilizando el enfoque entidad-relación (ER), identificando entidades, atributos y relaciones, según los requisitos del negocio. - Evalúa y ajusta el sistema de base de datos durante su ciclo de vida, incluyendo la revisión periódica del diseño y la aplicación de mejoras basadas en el feedback de los usuarios y el rendimiento del sistema, según los estándares de calidad de la empresa. - Evidencia pensamiento lógico y creatividad mientras desarrolla el diseño de un sistema de base de datos, según los principios de modelado de datos y los requerimientos del negocio.

Unidades de competencias técnicas	Elementos de competencia	Criterios de desempeño	Resultados de aprendizaje	Criterios de evaluación
3. Aplicar técnicas en el análisis y diseño de sistemas, utilizando las tecnologías y construyendo soluciones de software.	2.1 Manejar terminología orientada a objetos, de acuerdo a los estándares.	- Elaborar diagramas de clases utilizando la terminología y notación estándar de la POO, según modelado UML (Lenguaje Unificado de Modelado). - Desarrollar diagramas de secuencia y colaboración utilizando principios de orientación a objetos, según los estándares de diseño UML - Diseñar interfaces y clases abstractas para un sistema orientado a objetos, utilizando terminología y técnicas de POO, según las mejores prácticas de diseño orientado a objetos	1. Maneja terminología orientada a objetos, para elaborar y diseñar diferentes diagramas.	La persona participante: - Define los conceptos fundamentales de la Programación Orientada a Objetos (POO), según las definiciones estándares de POO. - Elabora diagramas de clases utilizando la terminología y notación estándar de la POO, según modelado UML (Lenguaje Unificado de Modelado). - Desarrolla diagramas de secuencia y colaboración utilizando principios de orientación a objetos, según los estándares de diseño UML - Diseña interfaces y clases abstractas para un sistema orientado a objetos, utilizando terminología y técnicas de POO, según las mejores prácticas de diseño orientado a objetos - Evidencia creatividad y responsabilidad mientras maneja terminología orientada a objetos, de acuerdo a los estándares.
	2.2 Gestionar el modelado de software y el proceso de desarrollo, según los estándares establecidos.	- Desarrollar diagramas de casos de uso utilizando UML, según las normas del modelado UML. - Evaluar los diagramas de casos de uso para reflejar adecuadamente los requisitos del sistema y las interacciones entre actores y casos de uso, según los principios de modelado de software. - Ajustar los diagramas de casos de uso para reflejar adecuadamente los requisitos del sistema y las interacciones entre actores y casos de uso, según los principios de modelado de software. - Diseñar la arquitectura del flujo de trabajo para análisis de requisitos y desarrollo de casos de uso, según los estándares de diseño de software.	2. Gestiona el modelado de software y el proceso de desarrollo, para crear diagramas de casos.	La persona participante: - Define concepto de modelado, según procedimientos. - Desarrolla diagramas de casos de uso utilizando UML, según las normas del modelado UML. - Evalúa y ajusta los diagramas de casos de uso para reflejar adecuadamente los requisitos del sistema y las interacciones entre actores y casos de uso, según los principios de modelado de software. - Diseña la arquitectura del flujo de trabajo para análisis de requisitos y desarrollo de casos de uso, según los estándares de diseño de software. - Evidencia responsabilidad y adhesión a las normas mientras gestiona el modelado de software y el proceso de desarrollo, según los estándares establecidos.

Unidades de competencias técnicas	Elementos de competencia	Criterios de desempeño	Resultados de aprendizaje	Criterios de evaluación
	2.3 Crear formularios de casos de uso con una planilla, de acuerdo a los estándares exigidos.	- Seleccionar escenarios de casos de uso, y elabora formularios prácticos para su análisis y desarrollo, según análisis de requisitos. - Grabar escenarios de casos de uso, y elabora formularios prácticos para su análisis y desarrollo, según análisis de requisitos. - Utilizar herramientas para la grabación y análisis de escenarios de casos de uso, aplicando técnicas de documentación y modelado, según los estándares de análisis y documentación de software. - Desarrollar un glosario de términos y especificaciones suplementarias para complementar la creación y comprensión de los casos de uso, según los requisitos del proyecto.	3. Crea formularios de casos de uso con una planilla, para crear diagramas de actividades.	La persona participante: - Identifica los diferentes elementos necesarios en un diagrama de actividad, según las técnicas de modelado UML. - Selecciona y graba escenarios de casos de uso, y elabora formularios prácticos para su análisis y desarrollo, según análisis de requisitos. - Utiliza herramientas para la grabación y análisis de escenarios de casos de uso, aplicando técnicas de documentación y modelado, según los estándares de análisis y documentación de software. - Desarrolla un glosario de términos y especificaciones suplementarias para complementar la creación y comprensión de los casos de uso, según los requisitos del proyecto. - Evidencia creatividad y atención al detalle mientras crea formularios de casos de uso con una planilla, de acuerdo a los estándares exigidos.
	2.4 Manejar las abstracciones claves, de acuerdo a los estándares exigidos.	- Utilizar el análisis CRC (Clases, Responsabilidades, Colaboradores) para seleccionar una abstracción candidata clave, según las metodologías de diseño de software. - Aplicar técnicas de identificación de abstracciones claves en la selección de los candidatos más relevantes para el desarrollo de casos de uso, según los principios de diseño orientado a objetos. - Actualizar y ajustar las abstracciones candidatas clave, según los cambios en los requisitos del proyecto y el análisis continuo.	4. Maneja las abstracciones claves, para identificar casos relevantes.	La persona participante: - Identifica las posibles abstracciones clave utilizando herramientas y técnicas adecuadas, según los principios de análisis de abstracciones. - Utiliza el análisis CRC (Clases, Responsabilidades, Colaboradores) para seleccionar una abstracción candidata clave, según las metodologías de diseño de software. - Aplica técnicas de identificación de abstracciones claves en la selección de los candidatos más relevantes para el desarrollo de casos de uso, según los principios de diseño orientado a objetos. - Actualiza y ajusta las abstracciones candidatas clave, según los cambios en los requisitos del proyecto y el análisis continuo. - Evidencia coherencia y responsabilidad mientras maneja las abstracciones claves, de acuerdo a los estándares exigidos.

Unidades de competencias técnicas	Elementos de competencia	Criterios de desempeño	Resultados de aprendizaje	Criterios de evaluación
	2.5 Construir modelo de ámbito o dominio del problema, según las técnicas de modelado de software establecidas.	- Dibujar flechas de navegación en el modelo de dominio para indicar las relaciones y direcciones entre clases, según las normas de diagramación UML. - Crear un modelo de dominio dibujando nodos de clase, estableciendo asociaciones, etiquetando las asociaciones, nombrando roles y definiendo multiplicidades, según los pasos establecidos para la construcción de modelos de dominio. - Realizar la validación del modelo de dominio utilizando diagramas de objetos para verificar la corrección y coherencia del modelo de clases, según las prácticas de verificación de modelos de software.	5. Construye modelo de ámbito o dominio del problema, para validar el tipo de dominio de una organización.	La persona participante: - Identifica los elementos esenciales de un diagrama de clases, según los principios de modelado UML. - Dibuja flechas de navegación en el modelo de dominio para indicar las relaciones y direcciones entre clases, según las normas de diagramación UML. - Crea un modelo de dominio dibujando nodos de clase, estableciendo asociaciones, etiquetando las asociaciones, nombrando roles y definiendo multiplicidades, según los pasos establecidos para la construcción de modelos de dominio. - Realiza la validación del modelo de dominio utilizando diagramas de objetos para verificar la corrección y coherencia del modelo de clases, según las prácticas de verificación de modelos de software. - Evidencia organización y resolución de problema mientras construye modelo de ámbito o dominio del problema, según las técnicas de modelado de software establecidas.
	2.6 Construir la solución de software, de acuerdo a los procesos técnicos.	- Aplicar principios de encapsulación y selección del tipo de datos, según los principios de diseño orientado a objetos. - Elaborar atributos derivados y perfecciona los metadatos del modelo de dominio, según los principios de diseño orientado a objetos. - Desarrollar métodos de relación muchos-a-muchas en el modelo de dominio, según las técnicas de refinación de diseño.	6. Construye la solución de software, para refinar el diseño de clase y realizar prueba	La persona participante: - Identifica diferentes atributos del modelo de dominio , según procesos técnicos. - Aplica principios de encapsulación y selección del tipo de datos, según los principios de diseño orientado a objetos. - Elabora atributos derivados y perfecciona los metadatos del modelo de dominio, según los principios de diseño orientado a objetos. - Desarrolla métodos de relación muchos-a-muchas en el modelo de dominio, según las técnicas de refinación de diseño. - Evidencia trabajo en equipo y disciplina mientras construye la solución de software, de acuerdo a los procesos técnicos.

Unidades de competencias técnicas	Elementos de competencia	Criterios de desempeño	Resultados de aprendizaje	Criterios de evaluación
4. Diseñar y manejar bases de datos.	3.1 Diseñar sistema de base de datos, según los requerimientos exigidos.	- Desarrollar el diseño conceptual de la base de datos, según procedimientos. - Optimizar el proceso de diseño de la base de datos, según las necesidades del sistema. - Desarrollar un plan de implementación y ajuste continuo del sistema de base de datos, según procedimientos.	1. Diseña sistema de base de datos, para ser implementado y ajustado.	La persona participante: - Identifica los diferentes tipos de usuarios y actores (roles) en un escenario de base de datos, según el diseño del sistema. - Desarrolla el diseño conceptual de la base de datos, según procedimientos. - Optimiza el proceso de diseño de la base de datos, según las necesidades del sistema. - Desarrolla un plan de implementación y ajuste continuo del sistema de base de datos, según procedimientos. - Demuestra responsabilidad e iniciativa mientras diseña sistema de base de datos, según los requerimientos exigidos.
	3.2 Diseñar un esquema relacional, según requerimientos y procesos exigidos.	- Aplicar la regla de transformación para convertir entidades y entidades débiles del modelo ER a tablas relacionales, según los pasos del algoritmo de transformación. - Crear vistas simples y complejas en SQL para la recuperación eficiente de datos, según los requisitos de las consultas - Implementar la transformación de relaciones (1:1, 1, M) del modelo ER a un esquema relacional, según las técnicas de normalización establecidas.	2. Diseña un esquema relacional, para la creación de tablas en una base de datos utilizando aplicaciones SQL.	La persona participante: - Identifica las reglas de transformación de un modelo entidad-relación (ER) a un modelo relacional, según los principios establecidos. - Aplica la regla de transformación para convertir entidades y entidades débiles del modelo ER a tablas relacionales, según los pasos del algoritmo de transformación. - Crea vistas simples y complejas en SQL para la recuperación eficiente de datos, según los requisitos de las consultas - Implementa la transformación de relaciones (1:1, 1, M) del modelo ER a un esquema relacional, según las técnicas de normalización establecidas. - Evidencia responsabilidad mientras diseña un esquema relacional, según requerimientos y procesos exigidos.

Unidades de competencias técnicas	Elementos de competencia	Criterios de desempeño	Resultados de aprendizaje	Criterios de evaluación
	3.3 Realizar consultas y subconsultas, según las necesidades del sistema de gestión de bases de datos.	- Realizar subconsultas en SQL para recuperar información de múltiples tablas, asegurando la correcta integración de datos entre diferentes fuentes, Según las técnicas de consulta SQL. - Desarrollar índices en SQL para optimizar el rendimiento de las consultas, basándose en el análisis de consultas y transacciones, según las decisiones de diseño de índices. - Diseñar consultas SQL avanzadas para realizar operaciones complejas en la base de datos, utilizando subconsultas, funciones agregadas, y cláusulas de filtrado, Según las necesidades de procesos específicos en SQL. - Ejecutar consultas SQL avanzadas para realizar operaciones complejas en la base de datos, utilizando subconsultas, funciones agregadas, y cláusulas de filtrado, Según las necesidades de procesos específicos en SQL.	3. Realiza consultas y subconsultas, para aplicar diferentes procesos en SQL.	La persona participante: - Identifica los diferentes factores que influyen en el diseño físico de una base de datos, según instrucciones establecidas. - Realiza subconsultas en SQL para recuperar información de múltiples tablas, asegurando la correcta integración de datos entre diferentes fuentes, Según las técnicas de consulta SQL. - Desarrolla índices en SQL para optimizar el rendimiento de las consultas, basándose en el análisis de consultas y transacciones, según las decisiones de diseño de índices. - Diseña y ejecuta consultas SQL avanzadas para realizar operaciones complejas en la base de datos, utilizando subconsultas, funciones agregadas, y cláusulas de filtrado, Según las necesidades de procesos específicos en SQL. - Evidencia trabajo en equipo y resolución de problemas mientras realiza consultas y subconsultas, según las necesidades del sistema de gestión de bases de datos.

Unidades de competencias técnicas	Elementos de competencia	Criterios de desempeño	Resultados de aprendizaje	Criterios de evaluación
5. Programar en lenguaje C++.	4.1 Crear variables y conversión de datos, según normativas establecidas.	- Declarar y utilizar variables y constantes, según las prácticas de programación orientada a objetos. - Crear y manejar estructuras de control de selección e iteración, incluyendo if simple, if-else, if-else if, switch, for, while, y estructuras anidadas, según las normas de control de flujo en programación. - Implementar y manejar métodos y funciones en Objective-C++, incluyendo el método principal main, para modularizar el código y definir la funcionalidad principal del programa, según los principios de diseño de funciones y métodos.	1. Crea variables y conversión de datos, para almacenar archivos y directorios desde objective C++.	La persona participante: - Explica el proceso de creación de variables y la conversión de datos, según procedimientos e instrucciones. - Declara y utiliza variables y constantes, según las prácticas de programación orientada a objetos. - Crea y maneja estructuras de control de selección e iteración, incluyendo if simple, if-else, if-else if, switch, for, while, y estructuras anidadas, según las normas de control de flujo en programación. - Implementa y maneja métodos y funciones en Objective-C++, incluyendo el método principal main, para modularizar el código y definir la funcionalidad principal del programa, según los principios de diseño de funciones y métodos. - Evidencia pensamiento lógico y atención al detalle mientras crea variables y conversión de datos, según normativas establecidas.
	4.2 Crear arreglos, según procedimientos exigidos.	- Declarar y utilizar arreglos unidimensionales, bidimensionales y multidimensionales, según las técnicas de manejo de datos en programación. - Eliminar elementos de un arreglo, según las prácticas de gestión dinámica de datos. - Implementar el acceso directo a archivos indexados mediante el uso de arreglos, según los principios de optimización y eficiencia en el manejo de datos.	2. Crea arreglos, para insertar y añadir elementos.	La persona participante: - Explica el proceso de creación e inserción de arreglos, según explicaciones dadas. - Declara y utiliza arreglos unidimensionales, bidimensionales y multidimensionales, según las técnicas de manejo de datos en programación. - Elimina elementos de un arreglo, según las prácticas de gestión dinámica de datos. - Implementa el acceso directo a archivos indexados mediante el uso de arreglos, según los principios de optimización y eficiencia en el manejo de datos. - Demuestra orden y creatividad mientras crea arreglos, según procedimientos exigidos.

Unidades de competencias técnicas	Elementos de competencia	Criterios de desempeño	Resultados de aprendizaje	Criterios de evaluación
	4.3 Ejecutar diferentes funciones de recursividad y sobrecarga, según procesos establecidos.	- Declarar y utilizar estructuras de selección (if, if-else, switch) y estructuras de iteración (for, while, do-while), según las mejores prácticas de programación. - Utilizar prototipos de funciones para declarar y definir funciones con diferentes tipos de parámetros y retornos, según las prácticas de diseño de software. - Aplicar la sobrecarga de métodos y diferencia entre recursividad e iteración, según las necesidades del problema.	3. Ejecuta diferentes funciones de recursividad y sobrecarga, para realizar declaraciones e interacciones.	La persona participante: - Analiza las funciones y la aplicación de recursividad y sobrecarga, según procedimientos. - Declara y utiliza estructuras de selección (if, if-else, switch) y estructuras de iteración (for, while, do-while), según las mejores prácticas de programación. - Utiliza prototipos de funciones para declarar y definir funciones con diferentes tipos de parámetros y retornos, según las prácticas de diseño de software. - Aplica la sobrecarga de métodos y diferencia entre recursividad e iteración, según las necesidades del problema. - Evidencia responsabilidad mientras ejecuta diferentes funciones de recursividad y sobrecarga, según procesos establecidos.
	4.4 Gestionar funciones virtuales, según requerimientos exigidos.	- Declarar y utilizar operadores y objetos grandes, y aplica asignación e inicialización en C++, según requerimientos exigidos. - Desarrollar y gestionar excepciones en C++, aplicando tratamiento de excepciones con throw, catch y try, según los procesos técnicos. - Gestionar funciones virtuales, incluyendo métodos puros y abstractos en C++, para permitir el polimorfismo y la implementación de clases abstractas, según requerimientos.	4. Gestiona funciones virtuales, para manejar métodos puros y abstractos.	La persona participante: - Define concepto de sobrecarga, según instrucciones recibidas. - Declara y utiliza operadores y objetos grandes, y aplica asignación e inicialización en C++, según requerimientos exigidos. - Desarrolla y gestiona excepciones en C++, aplicando tratamiento de excepciones con throw, catch y try, según los procesos técnicos. - Gestiona funciones virtuales, incluyendo métodos puros y abstractos en C++, para permitir el polimorfismo y la implementación de clases abstractas, según requerimientos. - Evidencia gestión de tiempo y adhesión a las normas mientras gestiona funciones virtuales, según requerimientos exigidos.

Unidades de competencias técnicas	Elementos de competencia	Criterios de desempeño	Resultados de aprendizaje	Criterios de evaluación
	4.5 Crear clases, según requerimientos exigidos.	- Diseñar y documentar librerías de clases en C++, según principios de estandarización. - Desarrollar e implementar clases contenedor y clases para aplicaciones en C++, optimizando la gestión de memoria, según los estándares establecidos. - Crear e implementar interfaces e implementaciones en C++, incluyendo clases de interfaz, según las directrices de arquitectura de software.	5. Crea clases, para diseñar bibliotecas.	La persona participante: - Explica el proceso de creación y jerarquías de clases y el diseño de bibliotecas, según instrucciones dadas. - Diseña y documenta librerías de clases en C++, según principios de estandarización. - Desarrolla e implementa clases contenedor y clases para aplicaciones en C++, optimizando la gestión de memoria, según los estándares establecidos. - Crea e implementa interfaces e implementaciones en C++, incluyendo clases de interfaz, según las directrices de arquitectura de software. - Evidencia pensamientos lógico y atención al detalle mientras crea clases, según requerimientos exigidos.

Unidades de competencias técnicas	Elementos de competencia	Criterios de desempeño	Resultados de aprendizaje	Criterios de evaluación
6. Desarrollar aplicaciones de software utilizando el lenguaje de programación C#, aplicando principios de programación orientada a objetos, estructuras de control, manejo de excepciones y acceso a bases de datos.	5.1 Crear formularios dentro del entorno gráfico.	- Ajustar el entorno de desarrollo seleccionando el namespace y organizando los assembly, según los requisitos del proyecto. - Diseñar formularios modales y no modales, según los requerimientos del usuario. - Desarrollar módulos estándar y módulos de formulario, según los principios de modularidad.	1. Crea formularios dentro del entorno gráfico, para desarrollar proyecto.	La persona participante: - Define concepto de objetos y propiedades de un formulario, según requerimientos de funcionalidad. - Ajusta el entorno de desarrollo seleccionando el namespace y organizando los assembly, según los requisitos del proyecto. - Diseña formularios modales y no modales, según los requerimientos del usuario. - Desarrolla módulos estándar y módulos de formulario, según los principios de modularidad. - Evidencia creatividad y coherencia mientras crea formularios dentro del entorno gráfico.
	5.2 Manejar la programación orientada a objetos, según las especificaciones técnicas.	- Desarrollar clases con propiedades y métodos, según los requerimientos de funcionalidad. - Implementar el encapsulamiento y la ocultación de datos, de acuerdo a las propiedades y métodos críticos. - Crear nuevas clases aplicando herencia, según las especificaciones de diseño.	2. Maneja la programación orientada a objetos, para aplicar clases, métodos y propiedades técnicas.	La persona participante: - Define concepto de POO, según procedimientos. - Desarrolla clases con propiedades y métodos, según los requerimientos de funcionalidad. - Implementa el encapsulamiento y la ocultación de datos, de acuerdo a las propiedades y métodos críticos. - Crea nuevas clases aplicando herencia, según las especificaciones de diseño. - Evidencia pensamiento lógico y adhesión a las normas mientras maneja la programación orientada a objetos, según las especificaciones técnicas.
	5.3 Integrar y configurar controles estándar, avanzados y adicionales, según especificaciones establecidas.	- Añadir y ajustar controles estándar como Label, TextBox, Button, y Checkbox en el formulario, según los requisitos del proyecto. - Implementar y personalizar controles avanzados como ListBox, ComboBox, HScrollBar, VScrollBar, y Image, según las necesidades de navegación y visualización de datos. - Implementar y utilizar controles de datos como DBGrid, MSFlexGrid, y DBCombo, según las especificaciones del proyecto.	3. Integra y configura controles estándar, avanzados y adicionales, para la gestión de datos y la interacción con el usuario en un proyecto.	La persona participante: - Define concepto de casilla de control de verificación, según procedimientos. - Añade y ajusta controles estándar como Label, TextBox, Button, y Checkbox en el formulario, según los requisitos del proyecto. - Implementa y personaliza controles avanzados como ListBox, ComboBox, HScrollBar, VScrollBar, y Image, según las necesidades de navegación y visualización de datos. - Implementa y utiliza controles de datos como DBGrid, MSFlexGrid, y DBCombo, según las especificaciones del proyecto. - Evidencia responsabilidad y adhesión las normas mientras integra y configura controles estándar, avanzados y adicionales, según especificaciones establecidas.

Unidades de competencias técnicas	Elementos de competencia	Criterios de desempeño	Resultados de aprendizaje	Criterios de evaluación
	5.4 Utilizar herramientas depuración, validación de datos y manipulación de errores, según procedimientos.	- Implementar técnicas de validación de datos a nivel de formulario, según los requisitos del proyecto. - Utilizar herramientas de inspección y ventanas de depuración para análisis detallado, según las especificaciones establecidas. - Configurar el proceso de manejo de errores, según las necesidades del sistema.	4. Utiliza herramientas depuración, validación de datos y manipulación de errores, para ejecutar el sistema de depuración, validación de datos y manipulación de errores.	La persona participante: - Explica el proceso de manipulación de errores, según procedimientos. - Implementa técnicas de validación de datos a nivel de formulario, según los requisitos del proyecto. - Utiliza herramientas de inspección y ventanas de depuración para análisis detallado, según las especificaciones establecidas. - Configura el proceso de manejo de errores, según las necesidades del sistema. - Evidencia responsabilidad y adhesión a las normas mientras utiliza herramientas depuración, validación de datos y manipulación de errores, según procedimientos.

3.3 ENTORNO PROFESIONAL:

- El profesional de esta cualificación desarrolla su actividad profesional en Programación básica de computadora:
 - Empresas desarrolladoras de software.
 - Todo tipo de empresas grandes, medianas y pequeñas con software y redes instalados.
- Sector Productivo: desarrolla sus actividades
 - Desarrolla su actividad en la economía del sector servicios a nivel público o privado o a título personal.
- OCUPACIONES O PUESTOS DE TRABAJOS RELEVANTES:
- Código CNO-2019: 2513 Grupo Primario: Desarrolladores Web y multimedia.
 - Desarrollador de internet
 - Desarrollador de sitios Web
 - Programador de animaciones
 - Programador de juegos de computador

4 Formación asociada al perfil.

NOMBRE DE LA CUALIFICACIÓN

Programación básica de computadora

No.	UNIDADES DE COMPETENCIA	MÓDULOS DE APRENDIZAJE	TIPO DE MÓDULO	CARGA HORARIA	DISTRIBUCIÓN HORARIA/MODALIDAD	
					V	P
1.	UCE_INCO_0001_CAP.; UCE_INCO_0002_CAP.; UCE_INCO_0003_CAP.; UCE_INCO_0004_CAP.; UCE_INCO_0005_CAP.; UCE_INCO_0006_CAP:	Matemática discreta	Básica	30 horas	09 horas	21 horas
2.	UCE_INCO_0001_CAP: Instalar sistemas operativos realizando tareas comunes.	Introducción al computador y sistema Operativo	Técnico	30 horas	09 horas	21 horas
3.	UCE_INCO_0002_CAP: Manejar lógica de programación, utilizando algoritmos y elaborando diagrama de flujo.	Introducción a la programación	Técnico	55 horas	16 horas	39 horas
4.	UCE_INCO_0003_CAP: Aplicar técnicas en el análisis y diseño de sistemas, utilizando las tecnologías y construyendo soluciones de software.	Análisis y Diseño de Sistemas I	Técnico	55 horas	16 horas	39 horas
5.	UCE_INCO_0004_CAP: Diseñar y manejar bases de datos.	Introducción a Base de Datos.	Técnico	55 horas	16 horas	39 horas
6.	UCE_INCO_0005_CAP: Programar en lenguaje C++.	Programación en Lenguaje C++	Técnico	50 horas	15 horas	35 horas
	UCE_INCO_0006_CAP: Desarrollar aplicaciones de software utilizando el lenguaje de programación C#, aplicando principios de programación orientada a objetos, estructuras de control, manejo de excepciones y acceso a bases de datos.	Programación C#	Técnico	60 horas	18 horas	42 horas
Total, de horas				335 horas	99 horas	236 horas

ESTRATEGIAS PARA EJECUCIÓN DEL PROGRAMA DE FORMACIÓN

Nota 1: Los módulos de aprendizaje en un programa de formación se presentan en un orden numérico secuencial, no obstante, los módulos básicos y transversales (de aquellos programas que lo poseen) podrán desarrollarse de manera simultánea con los módulos técnicos, estos últimos, sí deben mantener el orden que está señalado en el programa.

Nota 2: Los módulos transversales genéricos tipos MOOC (Cursos Masivos Abiertos en Línea) que se ejecutaran a través de la plataforma de Infotep Virtual, son módulos obligatorios para obtener el título de Técnico, estos podrán desarrollarse de manera simultánea o antes de iniciar con los módulos técnicos. Por tal motivo, a la hora de planificar el programa de formación, se deben contemplar para su ejecución.

- Formación humana.
- Protección al medio ambiente y cambio climático.

Nota 3: La carga horaria establecida en los módulos técnicos, podrá ser redistribuida por el facilitador partiendo de los conocimientos previos que sobre el área tengan los participantes, las necesidades o características del grupo y según el avance que muestren los mismos en el logro de los resultados. Esta acción deberá justificarla dando muestra de evidencias de la redistribución y el porqué de las mismas. Esta redistribución no debe afectar la carga horaria total del programa.

Nota 4: Para el desarrollo de los módulos básicos, transversales y técnicos, podrán ser ejecutados de acuerdo a la distribución de horas sugerida en la Formación asociada al perfil o a criterio de la instancia correspondiente y según aplique: V (virtual), P (presencial) o híbrida.

Nota 5: Para la ejecución de este programa se debe utilizar la estrategia de Aprendizaje Basado en Proyectos (ABP). El docente guiará al participante al desarrollo de un proyecto integrado que recoja las competencias de los módulos. Para ello es obligatorio que el docente haya recibido entrenamiento en la metodología de ABP.

- Para facilitar el desarrollo de los proyectos se colocará en la plataforma virtual un curso de autogestión de Aplicación de la estrategia de Aprendizaje basado en Proyecto (ABP), el cual debe ser desarrollado por los participantes, de modo que puedan obtener los conocimientos e instrucciones sobre cómo ejecutar el proyecto en todas sus fases.

Nota 6: Norma técnica de competencia laboral (NTCL); UC: Unidad de competencia.

5 DESCRIPCIÓN DE LOS MÓDULOS DE APRENDIZAJE
MÓDULO DE APRENDIZAJE No. 1

Código del Módulo	DENOMINACIÓN DEL MÓDULO DE APRENDIZAJE.				
	Matemática discreta				
Correspondencia con la unidad de competencia:	UCE_INCO_0001_TEC; UCE_INCO_0002_TEC; UCE_INCO_0003_TEC; UCE_INCO_0004_TEC; UCE_INCO_0005_TEC;				
Competencia final del módulo:	Al finalizar el módulo de aprendizaje, las personas participantes estarán en capacidad desarrollar diferentes operaciones algebraicas a través de análisis y posibilidades lógicas, bajo supervisión con responsabilidad sobre el trabajo propio.				
DURACIÓN EN HORAS:	30 horas.	HORAS TEÓRICAS:	09 horas.	HORAS PRÁCTICAS:	21 horas.
RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN	
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)		
1. Realiza operaciones entre conjuntos, para demostrar diversos teoremas relacionados con los números enteros.	- Cantidades numéricas. - Números enteros. - La adición. - La sustracción. - La multiplicación. - La división. - Número decimal. - Definición y partes. - Interpretación de números. - Procedimiento para cambiar fracción decimal en escritura decimal. - Procedimientos para sumar números decimales. - Números fraccionarios. - Definición y partes. - Clasificación. - Conversión de fracciones en decimal. - Simplificaciones de fracciones. - Reducción de fracciones con distintos denominadores. - Adición de fracciones en diferentes formas. - Multiplicación de fracciones. - División de fracciones. - Procedimientos para resolver problemas con fracciones.	- Aplica operaciones fundamentales (adición, sustracción, multiplicación, y división). - Realizar conversiones entre fracciones y números decimales. - Simplificar fracciones. - Reducir fracciones con distintos denominadores.	- Responsabilidad. - Escucha activa. - Participación. - Cooperación. - Iniciativa.	El participante: - Identifica los diferentes tipos de números (naturales, enteros, fraccionarios y decimales), según asignaciones. - Realiza conversiones entre fracciones y números decimales, según procedimientos. - Simplifica y reduce fracciones con distintos denominadores. - Evidencia responsabilidad mientras realiza operaciones entre conjuntos, según procedimientos requeridos.	

CONTINUACIÓN...1

RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)	
	- Sistema numérico decimal. - Símbolos y base exponente. - Valor de posiciones. - Sistema numérico binario o base 2. - Símbolos. - Potencias de 10. - Métodos de conversión. - Ejercicios de conversión de binario a decimal y de decimal a binario. - Representación en notación decimal. - Los octetos. - Números binarios de 32 bits. - Sistema numérico hexadecimal. - Símbolos. - Tabla de conversión. - Ejemplos de conversión de sistema binario, decimal y hexadecimal. - Lógica booleana o binaria. - Operaciones lógicas con AND, OR y NOT. - Conjuntos y subconjuntos. - Operaciones con conjuntos. - Sucesiones. - Divisiones en los enteros. - Estructuras matemáticas. - Algoritmos y pseudocódigos. - Diagrama de Venn. - Interpretación de la notación simbólica en la definición de conjuntos. - Propositiones y operaciones lógicas. - Sintaxis y la semántica de la lógica. - Conectivos lógicos y proposiciones compuestas.			-

CONTINUACIÓN...1

RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)	
	- Cuantificadores. - Propositiones condicionales. - Métodos de demostración. - Álgebra declarativa. - Reglas de inferencia. - Evaluación de expresiones. - Operaciones binarias. - Propiedades de las operaciones binarias. - Semigrupos: definición y propiedades. - Productos y Cocientes. - Grupos: definición y propiedades. - Productos y cocientes.			

CONTINUACIÓN...1

RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)	
2. Realiza operaciones de potenciación, para calcular potencia de base y exponente natural.	- Multiplicación de factores iguales. - Potencias cuya base sean números negativos y exponentes pares e impares. - Verificación de las propiedades distributivas de la potenciación con relación a la multiplicación y a la división de números enteros. - Potencia con base. - En número entero y como exponente el cero y la unidad. - Operaciones fundamentales de la potenciación. - Suma de potencias. - Resta de potencias. - División de potencias. - Multiplicación de potencias. - Concepto de radicación como operación inversa de la potenciación. - Elementos de un radical. - Índice de la raíz. - Radicando. - Raíz exacta. - Raíz inexacta. - Raíz de índice en número natural y radicando en número racional. - Suma, resta, multiplicación de números con radicales.	- Realizar operaciones de factores iguales. - Calcular potencia de base entera y exponente natural. - Realizar operaciones de números enteros y racionales con radicales.	- Responsabilidad. - Escucha activa. - Participación. - Cooperación. - Iniciativa.	El participante: - Socializa los diferentes tipos de dispositivos eléctricos, según procedimientos. - Realiza operaciones de factores iguales, según lo requerido. - Calcula potencia de base entera y exponente natural, según procedimientos exigidos. - Realiza operaciones de números enteros y racionales con radicales, conforme a requerimientos. - Evidencia participación mientras realiza operaciones de potenciación, según procedimientos establecidos.

CONTINUACIÓN...1

RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)	
3. Organiza ecuaciones algebraicas, para aplicar operaciones fundamentales en el análisis y solución de problemas matemáticos.	- Generalidades de algebra: • Expresiones algebraicas. • Ecuaciones algebraicas. • Clasificación de expresiones algebraicas de acuerdo a su: • Grado. • Orden. - Términos. - Operaciones fundamentales del álgebra: - Suma, resta, multiplicación, división, factorización, reducción de término semejantes. - Resolución de ecuaciones de primer grado. - Introducción al despeje de fórmulas. - Pasos para efectuar despeje de fórmulas. - Variables dependientes e independientes.	- Clasificar ecuaciones algebraicas, según lo requerido. - Realizar ecuaciones algebraicas. - Resolver ecuaciones de primer grado con una incógnita. - Realizar despeje de fórmulas algebraicas.	- Responsabilidad. - Escucha activa. - Participación. - Cooperación. - Iniciativa.	El participante: - Analiza sobre la clasificación y organización de ecuaciones algebraicas, según asignaciones. - Clasifica ecuaciones algebraicas, según lo requerido. - Realiza ecuaciones algebraicas, según requerimientos. - Resuelve ecuaciones de primer grado con una incógnita, conforme a lo requerido. - Realiza despeje de fórmulas algebraicas, según proceso y requerimientos. - Evidencia responsabilidad y disciplina mientras organiza ecuaciones algebraicas, según procedimientos requeridos.

Estrategias de enseñanza y aprendizaje

Las estrategias didácticas deberán centrarse en las personas participantes, favoreciendo que este gestione y construya activamente su aprendizaje. La misma deberá propiciar en el participante el trabajo colaborativo, comunicación efectiva, cierto nivel de autonomía y compromiso, el pensamiento crítico, la investigación, convivencia y resolución de problemas complejos y no rutinarios.

De igual manera, cada una de las estrategias utilizadas deberá permitir el empoderamiento de las personas participantes, donde desarrollen la capacidad de investigar, emitir opinión y tomar decisión con relación a la forma de solucionar los problemas planteados, según su mundo social y laboral. Todo esto implica la construcción de habilidades blandas, además de las de carácter técnico, y que este debe ir en correspondencia con lo que espera el sector productivo y la sociedad.

Las estrategias y actividades de enseñanza y aprendizaje recomendadas para contenidos de las áreas cognoscitivas y actitudinal.

Bitácora, edición entre pares, métodos de casos, panel expandido, argumento y refutación, Phillip 66, dramatizaciones, juego de roles, aprendizaje +Acción, aprendizaje basado en proyectos y otras estrategias que puedan contribuir al logro del aprendizaje.

Las estrategias y actividades de enseñanza y aprendizaje recomendadas para contenidos del área psicomotora.

Construyendo -aprendo, demostraciones ejecuciones, simulaciones, y otras estrategias que puedan contribuir al logro del aprendizaje.

De igual forma, el Facilitador podrá utilizar todas las estrategias que considere necesarias para asegurar el logro de las competencias según el programa.

Todas las estrategias deben ser congruentes y pertinentes con el contenido y los resultados de aprendizaje y deberán incluir:

Actividades individuales, en equipo, en línea y en el contexto real de trabajo, entre otros.

Nota: Durante el desarrollo o impartición de los módulos que comprenden un programa de formación, el docente tendrá la autonomía de impartir contenidos necesarios para lograr completar los resultados de aprendizaje que sean innovadores según la tendencia, actualidad, relacionados con la acción formativa que imparte.

El facilitador tendrá que dejar evidencias de la impartición de estos contenidos, registrando la temática impartida, en el programa de formación en el punto destinado a Sugerencias de Mejora para la Revisión del Programa de Formación.

MÓDULO DE APRENDIZAJE No. 2

Código del Módulo	DENOMINACIÓN DEL MÓDULO DE APRENDIZAJE				
	Introducción a la computadora y sistema operativo				
Correspondencia con la unidad de competencia:	UCE_INCO_0001_TEC:				
Competencia final del módulo:	Al finalizar el módulo de aprendizaje, las personas participantes estarán en capacidad de instalar sistemas operativos realizando tareas comunes, bajo supervisión con responsabilidad sobre el trabajo propio.				
DURACIÓN EN HORAS:	30 horas.	HORAS TEÓRICAS:	09 horas.	HORAS PRÁCTICAS:	21 horas.
RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN	
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)		
1. Configura y mantiene los componentes de hardware de un sistema de computación, para diagnosticar fallas y realizar el mantenimiento preventivo y correctivo.	- Antecedentes históricos. - Primeros inventos: - Aritmómetro de Pascal. • La máquina diferencial. • La máquina sumadora y totalizadora. - Primera computadora electromecánica. - Generación de las computadoras. - Conceptos: • Sistema de computación. • Nombres, funciones y características de los gabinetes y fuentes de energía. • Nombres, funciones y características de los componentes internos. - Nombres, funciones y características de las motherboards. - Nombres, funciones y características de los procesadores/CPU - Nombres, funciones y características de los sistemas de enfriamiento. - Nombres, funciones y características de ROM y RAM. - Nombres, funciones y características de las tarjetas adaptadoras.	- Configurar las motherboards, procesadores y módulos de memoria (RAM y ROM). - Configurar los puertos de comunicación (USB, HDMI, Ethernet). - Utilizar herramientas de diagnóstico para identificar fallas en los sistemas de enfriamiento y las tarjetas adaptadoras. - Evaluar el estado de las tarjetas de video y otros dispositivos gráficos. - Realizar el mantenimiento preventivo de las unidades de almacenamiento, incluyendo la limpieza física y la optimización del software, - Desarrollar y ejecutar un plan de mantenimiento preventivo periódico que incluye la revisión de todos los componentes de hardware.	- Precisión. - Orden. - Creatividad. - Organización. - Trabajo en equipo. - Responsabilidad. - Comunicación efectiva. - Orientación a la calidad y al detalle. - Pensamiento crítico. - Resolución de problema. - Gestión del tiempo. - Confidencialidad. - Adhesión a las normas.	El participante: - Define concepto de sistema de computación, según explicaciones dadas. - Configura las motherboards, procesadores y módulos de memoria (RAM y ROM)., según especificaciones técnicas. - Utiliza herramientas de diagnóstico para identificar fallas en los sistemas de enfriamiento y las tarjetas adaptadoras, siguiendo especificaciones técnicas. - Desarrolla y ejecuta un plan de mantenimiento preventivo periódico que incluye la revisión de todos los componentes de hardware, según procedimientos establecidos. - Evidencia responsabilidad y adhesión a las normas mientras configura y mantiene los componentes de hardware de un sistema de computación, según exigencias requeridas.	

CONTINUACIÓN...2

RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN
	SABER (CONCEPTOS)	SABER HACER (PROCEDIMIENTOS)	SABER SER (ACTITUDES)	
	- Nombres, funciones y características de las unidades de almacenamiento. - Nombres, funciones y características de los cables internos. - Nombres, funciones y características de los puertos y cables. - Nombres, funciones y características de los dispositivos de entrada y salida. - Explicar los recursos del sistema y su función, IRQ, dirección I/O y DMA. - Hardware. - CPU. - Memoria. - Procesadores. - Motherboard. - Switch de power. - Diskette. - CD-ROM. - DVD-ROM /BLUE RAY. - Zip Driver. - CD/DVD /BLUE RAY - Write. - Tarjetas. - Tarjetas de Video. - Modems. - Puertos paralelos. - Puertos seriales. - Puertos USB. - Escáner. - Impresoras (tipos). - Teclados (zonas {teclas de función, alfanuméricas, numéricas etc.}). - Mouse (tipos). - Bit. - Byte.			

CONTINUACIÓN...2

RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN
	SABER (CONCEPTOS)	SABER HACER (PROCEDIMIENTOS)	SABER SER (ACTITUDES)	
	- Kilobyte. - Megabyte. - Gigabyte. - Terabyte. - Peta byte. - Archivo. - Datos. - Información. - La función de las condiciones y procedimientos de trabajo seguro. - Procedimientos de seguridad y peligros potenciales para los usuarios y técnicos. - Procedimientos de seguridad para evitar el daño al equipo y la pérdida de datos. - Las herramientas y software utilizados con los componentes de la computadora personal y sus funciones. - Las herramientas de hardware y sus funciones. - Las herramientas de software y sus funciones. - Las herramientas organizacionales y sus funciones.			

CONTINUACIÓN...2

RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)	
2. Instala sistemas operativos, para manejar procedimientos de selección, instalación, configuración y soporte.	- Función y conceptos de un sistema operativo. - Características de los sistemas operativos modernos, incluyendo los de red y escritorio. - Las aplicaciones y los entornos que son compatibles con un sistema operativo. - Instalación personalizada. - Los archivos de secuencia de inicio y archivos de registro. - Manipulación de los archivos del sistema operativo.	- Realizar la instalación personalizada de un sistema operativo, seleccionando adecuadamente las particiones del disco y configurando opciones avanzadas. - Configurar los archivos de secuencia de inicio y archivos de registro del sistema operativo para optimizar el tiempo de arranque. - Desarrollar y aplicar procedimientos de soporte técnico post-instalación para asegurar la correcta operación del sistema operativo.	- Precisión. - Orden. - Creatividad. - Organización. - Trabajo en equipo. - Responsabilidad. - Comunicación efectiva. - Orientación a la calidad y al detalle. - Pensamiento crítico. - Resolución de problema. - Gestión del tiempo. - Confidencialidad. - Adhesión a las normas.	El participante: - Identifica las características de los sistemas operativos modernos, incluyendo los de red y de escritorio, según requerimientos. - Realiza la instalación personalizada de un sistema operativo, seleccionando adecuadamente las particiones del disco y configurando opciones avanzadas según los requisitos del cliente. - Configura los archivos de secuencia de inicio y archivos de registro del sistema operativo para optimizar el tiempo de arranque, según especificaciones técnicas. - Desarrolla y aplica procedimientos de soporte técnico post-instalación para asegurar la correcta operación del sistema operativo, según especificaciones técnicas. - Evidencia responsabilidad mientras instala sistema operativo, según estándares de informática.

CONTINUACIÓN...2

RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)	
3. Configura y optimiza sistemas operativos Windows, Linux y Mac OS, para asegurar el correcto funcionamiento de los equipos y maximizar el rendimiento y la seguridad del sistema.	- Las estructuras de directorios. - Actualización de un sistema operativo. - Técnicas comunes de mantenimiento preventivo de los sistemas operativos. - Las interfaces de Windows, Linux y Mac OS. - Archivos principales del sistema de Windows, Linux y Mac OS, función y ubicación. - Procedimientos para la instalación de dispositivos de Hardware. - Procedimientos para la optimización del sistema operativo. - Conceptos: • Windows. - Características. - Entorno de Windows: • Escritorio. • Iconos principales: • My PC. • Papelera de reciclaje. • Barra de tarea. • Botón inicio. - Uso del Mouse: • Botones. • Clic izquierdo. • Clic derecho. • Scroll. - Uso del botón inicio (Start). - Ventanas. - Abrir ventana.	- Instalar el sistema operativo con los valores por defecto. - Realizar actualizaciones del sistema operativo. - Configurar las opciones de ratón y teclado en Windows, ajustando la velocidad, idioma y preferencias de los botones. - Utilizar el botón de inicio (Start) y administra las ventanas en Windows, maximizando, minimizando y organizando eficientemente las ventanas abiertas. - Programar una tarea. - Crear copia del disco duro. - Ejecutar actualizaciones del sistema operativo sin comprometer la integridad del sistema. - Crear un plan de mantenimiento preventivo. - Realizar la resolución de problemas de los sistemas operativos. - Optimizar el sistema operativo mediante la configuración avanzada de las opciones del panel de control y la barra de tareas en Windows. - Desarrollar y ejecutar un plan de optimización del sistema operativo, utilizando herramientas del sistema y software de terceros para mejorar el rendimiento general del equipo.	- Precisión. - Orden. - Creatividad. - Organización. - Trabajo en equipo. - Responsabilidad. - Comunicación efectiva. - Orientación a la calidad y al detalle. - Pensamiento crítico. - Resolución de problema. - Gestión del tiempo. - Confidencialidad. - Adhesión a las normas.	El participante: - Identifica correctamente las interfaces de Windows, Linux y Mac OS, de acuerdo a explicaciones. - Ejecuta actualizaciones del sistema operativo sin comprometer la integridad del sistema, según los procedimientos de seguridad establecidos. - Realiza la resolución de problemas de los sistemas operativos, según los protocolos de diagnóstico y reparación definido. - Realiza la instalación de dispositivos de hardware en sistemas operativos Windows, Linux y Mac OS, según lo requerido. - Desarrolla y ejecuta un plan de optimización del sistema operativo, utilizando herramientas del sistema y software de terceros para mejorar el rendimiento general del equipo, de acuerdo a especificaciones técnicas. - Evidencia orientación a la calidad y al detalle mientras configura y optimiza sistemas operativos Windows, Linux y Mac OS, según los estándares de calidad y procedimientos establecidos.

CONTINUACIÓN...2

RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)	
	- Partes de una ventana. - Botones de control. - Minimizar. - Maximizar y/o restaurar. - Cerrar. - Mover ventana. - Modo de visualizar iconos. - Panel de control. - Fecha y hora. - Cambiar hora. - Cambiar fecha. - Mouse. - Punteros. - Botones. - Velocidad. - Teclado. - Velocidad. - Idioma. - Pantalla. - Cambiar fondo. - Protector de pantalla. - Apariencia. - Configuración. - Barra de tareas y Menú Inicio.			

RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)	
4. Opera sistemas operativos y herramientas de gestión de archivos, para administrar y mantener el equipo de manera eficiente.	- Localización de archivos y carpetas. - Creación de archivos y carpetas. - Eliminación de archivos y carpetas. - Organización de iconos. - Modos de visualizar de iconos. - Propiedades de archivos y carpetas. - Selección elementos. - Cambio de nombre de elementos. - Búsqueda de archivos o carpetas. - Propiedades de disco. - Formato de discos duros. - Interfaz de comandos (cmd) de Windows y los comandos y utilidades más comunes, incluyendo la sintaxis y operadores. - Las utilidades más comunes para el manejo y la administración del sistema, los archivos y carpetas en Ms Windows y su utilización. - Los códigos de error más comunes en Windows e identificar los pasos necesarios para resolverlos. - Introducción a Unix. - Instalación de Unix. - Introducción a Gnu/Linux. - Instalación de Gnu/Linux. - Introducción a Mac OS. - Instalación de MacOS X. - Android.	- Localizar y administrar archivos y carpetas en los sistemas operativos Windows, Linux, y Mac OS. - Crear, eliminar y organizar archivos y carpetas en el Explorador de Windows y en la línea de comandos (cmd) de Windows. - Optimizar la interfaz de usuario de los sistemas operativos Windows y Mac OS, ajustando la visualización de iconos y personalizando el entorno de trabajo. - Realizar copias de seguridad y restauración de datos en Windows y Linux - Configurar y utilizar las utilidades de administración del sistema en Windows, Unix, Gnu/Linux y Mac OS.	- Precisión. - Orden. - Creatividad. - Organización. - Trabajo en equipo. - Responsabilidad. - Comunicación efectiva. - Orientación a la calidad y al detalle. - Pensamiento crítico. - Resolución de problema. - Gestión del tiempo. - Confidencialidad. - Adhesión a las normas.	El participante: - Identifica los elementos del explorador de Windows, de acuerdo a instrucciones. - Realiza copias de seguridad y restauración de datos en Windows y Linux, según procedimientos establecidos. - Configura y utiliza las utilidades de administración del sistema en Windows, Unix, Gnu/Linux y Mac OS, según las especificaciones de seguridad. - Crea, elimina, y organiza archivos y carpetas en el Explorador de Windows y en la línea de comandos (cmd) de Windows, según los protocolos de gestión de archivos. - Evidencia responsabilidad y adhesión a las normas mientras opera sistemas operativos y herramientas de gestión de archivos, según las políticas de seguridad.

Estrategias de enseñanza y aprendizaje

Las estrategias didácticas deberán centrarse en las personas participantes, favoreciendo que este gestione y construya activamente su aprendizaje. La misma deberá propiciar en el participante el trabajo colaborativo, cierto nivel de autonomía y compromiso, el pensamiento crítico, la investigación, convivencia y resolución de problemas ligeramente complejos.

De igual manera, cada una de las estrategias utilizadas deberá permitir el empoderamiento de las personas participantes, donde desarrollen la capacidad de investigar, emitir opinión y tomar decisión con relación a la forma de solucionar los problemas planteados, según su mundo social y laboral. Todo esto implica la construcción de habilidades blandas, además de las de carácter técnico, y que este debe ir en correspondencia con lo que espera el sector productivo y la sociedad.

Las estrategias y actividades de enseñanza y aprendizaje recomendadas para contenidos de las áreas cognoscitivas y actitudinal.

Investigo- documento y aprendo, peceras, debates activos, mapas mentales, comunidades de aprendizaje, Phillip 66, redes de palabras, aprendizaje basado en problema, el aprendizaje basado en proyectos y otras que puedan contribuir al logro del aprendizaje.

Las estrategias y actividades de enseñanza y aprendizaje recomendadas para contenidos del área psicomotora.

Demostración / ejecución, construyendo- aprendo, simulaciones y otras que puedan contribuir al logro del aprendizaje.

De igual forma, el Facilitador podrá utilizar todas las estrategias que considere necesarias para asegurar el logro de las competencias según el programa.

Todas las estrategias deben ser congruentes y pertinentes con el contenido y los resultados de aprendizaje y deberán incluir:

Actividades individuales, en equipo, en línea y en el contexto real de trabajo, entre otros.

Nota: Durante el desarrollo o impartición de los módulos que comprenden un programa de formación, el docente tendrá la autonomía de impartir contenidos necesarios para lograr completar los resultados de aprendizaje que sean innovadores según la tendencia, actualidad, relacionados con la acción formativa que imparte.

El facilitador tendrá que dejar evidencias de la impartición de estos contenidos, registrando la temática impartida, en el programa de formación en el punto destinado a Sugerencias de Mejora para la Revisión del Programa de Formación.

MÓDULO DE APRENDIZAJE No. 3

Código del Módulo	DENOMINACIÓN DEL MÓDULO DE APRENDIZAJE				
	Introducción a la programación				
Correspondencia con la unidad de competencia:	UCE_INCO_0002_TEC:				
Competencia final del módulo:	Al finalizar el módulo de aprendizaje, las personas participantes estarán en capacidad de manejar lógica de programación, utilizando algoritmos y elaborando diagrama de flujo, manejándose en un ambiente de confianza y participación.				
DURACIÓN EN HORAS:	55 horas	HORAS TEÓRICAS:	16 horas	HORAS PRÁCTICAS:	39 horas
RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN	
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)		
1. Desarrolla la lógica de programación utilizando los ciclos básicos del procesamiento de información, para aplicar principios de programación estructurada y el manejo eficiente de datos.	- Evolución histórica. - Concepto de Programación, programa. - Tipos de programas (objeto, fuente). - Ciclo básico del procesamiento de datos. - Tipos de ciclo de procesamiento de datos (básico y ampliado). - Algunos términos en el procesamiento de datos: - Bit. - Byte. - Carácter. - Campo. - Registro. - Archivos (tipos). - Partes de un archivo. - Modo de acceso a los archivos. - Variables y constantes. - Tipos de variables. - Operadores. - Clasificación de operadores. - UML. - Requerimiento de negocio. - Estructuras de control y ciclos repetitivos.	- Crear y manipular variables y constantes en programas básicos. - Seleccionar y utilizar adecuadamente tipos de variables y constantes. - Desarrollar diagramas de flujo y modelos UML para representar estructuras de control y ciclos repetitivos en la programación. - Diseñar estructuras de control condicionales y ciclos repetitivos. - Codificar estructuras de control condicionales y ciclos repetitivos. - Utilizar diagramas UML para modelar requerimientos de negocio y estructuras de control.	- Precisión. - Orden. - Creatividad. - Organización. - Trabajo en equipo. - Responsabilidad. - Comunicación efectiva. - Orientación a la calidad y al detalle. - Pensamiento lógico. - Resolución de problema. - Gestión del tiempo. - Confidencialidad. - Adhesión a las normas.	El participante: - Explica la evolución histórica de la programación y su impacto en el desarrollo de software, según los principios básicos de procesamiento de información. - Selecciona y utiliza adecuadamente tipos de variables y constantes, según las necesidades de almacenamiento y procesamiento de datos. - Diseña y codifica estructuras de control condicionales y ciclos repetitivos, según los objetivos del programa. - Utiliza diagramas UML para modelar requerimientos de negocio y estructuras de control, según las especificaciones del cliente. - Evidencia pensamiento lógico y orientación a la calidad mientras desarrolla la lógica de programación utilizando los ciclos básicos del procesamiento de información, según los estándares de programación.	

RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)	
2. Desarrolla diagramas de flujo y algoritmos, para implementar soluciones eficientes en procesos informáticos.	- Estructura Si-Entonces-Sino. - Decisiones en cascada. - Decisiones en secuencia. - Decisiones anidadas. - Estructura CASE. - Estructura CASE simples. - Estructura CASE anidados. - Concepto general. - Tipos de ciclos. - Ciclo WHILE. - Ciclo FOR. - Ciclo DO UNTIL. - Ciclo DO WHILE. - Ciclos anidados. - Concepto general. - Índices. - Características. - Vectores. - Características. - Ejemplos de vectores. - Matrices.	- Diseñar diagramas de flujo utilizando la estructura Si-Entonces-Sino y decisiones en cascada. - Crear diagramas de flujo que representen ciclos de control, como WHILE, FOR, DO UNTIL y DO WHILE. - Desarrollar algoritmos que implementen estructuras de control, como decisiones en secuencia y decisiones anidadas. - Utilizar la estructura CASE para desarrollar algoritmos que manejen múltiples condiciones de forma eficiente. - Desarrollar algoritmos que utilicen vectores y matrices para almacenar y manipular datos. - Implementar ciclos anidados en algoritmos para resolver problemas complejos de manera eficiente. - Optimizar algoritmos mediante el uso adecuado de estructuras de control y ciclos anidados. - Corregir errores en los algoritmos y diagramas de flujo desarrollados, aplicando técnicas de depuración y pruebas.	- Precisión. - Orden. - Creatividad. - Organización. - Trabajo en equipo. - Responsabilidad. - Comunicación efectiva. - Orientación a la calidad y al detalle. - Pensamiento lógico. - Resolución de problema. - Gestión del tiempo. - Confidencialidad. - Adhesión a las normas.	El participante: - Analiza sobre elaboración de diagrama de flujo y algoritmos, según procedimientos. - Desarrolla algoritmos que utilicen vectores y matrices para almacenar y manipular datos, según las características del problema. - Optimiza algoritmos mediante el uso adecuado de estructuras de control y ciclos anidados, según los criterios de eficiencia y rendimiento del programa. - Corrige errores en los algoritmos y diagramas de flujo desarrollados, aplicando técnicas de depuración y pruebas, según los estándares de calidad del software - Evidencia pensamiento lógico y creatividad mientras desarrolla diagramas de flujo y algoritmos, según procedimientos.

RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)	
3. Desarrolla el diseño de un sistema de base de datos, para realizar su implementación y ajuste.	- Concepto de datos. - Concepto de base de datos. - Enfoque de bases de datos. - Clases de usuarios. - Actores (roles) en un escenario de Base de Datos. - Ventajas de un SGBD o motor de base de Datos. - Diseño conceptual de la base de datos. - Ciclo de vida de un sistema de información. - Ciclo de vida del sistema de aplicación de base de datos. - El proceso de diseño de base de datos. - Recopilación de información con los usuarios del negocio. - Técnicas de especificación de requisitos. - La necesidad de modelar los datos. - Modelado entidad relación ER. - Tipos de entidades, tipos de relaciones, atributos clave, restricciones de cardinalidad. - Identificación de las entidades, atributos y relaciones en un documento de requerimiento de negocios. - Costo de adquisición del software. - Costo de mantenimiento. - Costos de creación y conversión de la base de datos. - Costo de personal. - Costo de entrenamiento. - Costo de operación. - Algoritmo de transformación de ER en relacional.	- Recopilar información con los usuarios del negocio para especificar los requisitos de la base de datos, utilizando técnicas de especificación de requisitos adecuadas. - Diseñar el modelo conceptual de una base de datos utilizando el enfoque entidad-relación (ER), identificando entidades, atributos y relaciones, según los requisitos del negocio. - Desarrollar documentación detallada sobre el diseño de la base de datos - Elaborar un plan detallado que considere los costos asociados a la adquisición del software, mantenimiento, creación y conversión de la base de datos, personal, entrenamiento y operación. - Implementar el diseño de la base de datos en el SGBD seleccionado, ajustando la estructura según las necesidades específicas del negocio. - Evaluar el sistema de base de datos durante su ciclo de vida, incluyendo la revisión periódica del diseño y la aplicación de mejoras basadas en el feedback de los usuarios y el rendimiento del sistema. - Ajustar el sistema de base de datos durante su ciclo de vida, incluyendo la revisión periódica del diseño y la aplicación de mejoras basadas en el feedback de los usuarios y el rendimiento del sistema.	- Precisión. - Orden. - Creatividad. - Organización. - Trabajo en equipo. - Responsabilidad. - Comunicación efectiva. - Orientación a la calidad y al detalle. - Pensamiento lógico. - Resolución de problema. - Gestión del tiempo. - Confidencialidad. - Adhesión a las normas.	El participante: - Define concepto de datos y de bases de datos, según procedimientos. - Recopila información con los usuarios del negocio para especificar los requisitos de la base de datos, utilizando técnicas de especificación de requisitos adecuadas, según las necesidades del negocio. - Diseña el modelo conceptual de una base de datos utilizando el enfoque entidad-relación (ER), identificando entidades, atributos y relaciones, según los requisitos del negocio. - Evalúa y ajusta el sistema de base de datos durante su ciclo de vida, incluyendo la revisión periódica del diseño y la aplicación de mejoras basadas en el feedback de los usuarios y el rendimiento del sistema, según los estándares de calidad de la empresa. - Evidencia pensamiento lógico y creatividad mientras desarrolla el diseño de un sistema de base de datos, según los principios de modelado de datos y los requerimientos del negocio.

ESTRATEGIAS METODOLÓGICAS PARA EL PLANEAMIENTO DIDÁCTICO

Estrategias de enseñanza y aprendizaje

Las estrategias didácticas deberán centrarse en las personas participantes, favoreciendo que este gestione y construya activamente su aprendizaje. La misma deberá propiciar en el participante el trabajo colaborativo, cierto nivel de autonomía y compromiso, el pensamiento crítico, la investigación, convivencia y resolución de problemas ligeramente complejos.

De igual manera, cada una de las estrategias utilizadas deberá permitir el empoderamiento de las personas participantes, donde desarrollen la capacidad de investigar, emitir opinión y tomar decisión con relación a la forma de solucionar los problemas planteados, según su mundo social y laboral. Todo esto implica la construcción de habilidades blandas, además de las de carácter técnico, y que este debe ir en correspondencia con lo que espera el sector productivo y la sociedad.

Las estrategias y actividades de enseñanza y aprendizaje recomendadas para contenidos de las áreas cognoscitiva y actitudinal son:

Investigo- documento y aprendo, peceras, debates activos, mapas mentales, comunidades de aprendizaje, Phillip 66, redes de palabras, aprendizaje basado en problema, el aprendizaje basado en proyectos y otras que puedan contribuir al logro del aprendizaje.

Las estrategias y actividades de enseñanza y aprendizaje recomendadas para contenidos del área psicomotora son:

Demostración / ejecución, construyendo- aprendo, simulaciones y otras que puedan contribuir al logro del aprendizaje.

De igual forma, el Facilitador podrá utilizar todas las estrategias que considere necesarias para asegurar el logro de las competencias según el programa.

Todas las estrategias deben ser congruentes y pertinentes con el contenido y los resultados de aprendizaje y deberán incluir:

Actividades individuales, en equipo, en línea y en el contexto real de trabajo, entre otros.

Nota: Durante el desarrollo o impartición de los módulos que comprenden un programa de formación, el docente tendrá la autonomía de impartir contenidos necesarios para lograr completar los resultados de aprendizaje que sean innovadores según la tendencia, actualidad, relacionados con la acción formativa que imparte.

El facilitador tendrá que dejar evidencias de la impartición de estos contenidos, registrando la temática impartida, en el programa de formación en el punto destinado a Sugerencias de Mejora para la Revisión del Programa de Formación.

MÓDULO DE APRENDIZAJE No. 4

Código del Módulo	DENOMINACIÓN DEL MÓDULO DE APRENDIZAJE				
	Análisis y diseño de sistema I				
Correspondencia con la unidad de competencia:	UCE_INCO_0004_TEC:				
Competencia final del módulo:	Al finalizar el módulo de aprendizaje, las personas participantes estarán en la capacidad de aplicar técnicas en el análisis y diseño de sistemas, utilizando las tecnologías y construyendo soluciones de software, bajo supervisión con responsabilidad sobre el trabajo propio.				
DURACIÓN EN HORAS:	55 horas.	HORAS TEÓRICAS:	16 horas.	HORAS PRÁCTICAS:	39 horas.
RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN	
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)		
1. Maneja terminología orientada a objetos, para elaborar y diseñar diferentes diagramas.	- Examinando Orientación a Objetos. - La complejidad del software. - Descomposición Software. - Costos de Software. - La comparación de los paradigmas: Estructurado Vs Orientado a Objeto. - Conceptos Fundamentales OO: - Objetos. - Clases. - Abstracción. - La encapsulación. - Herencia. - Clases abstractas. - Interfaces. - Polimorfismo. - Cohesión. - Coupling. - Las asociaciones de clase y los enlaces de objeto. - Delegación.	- Elaborar diagramas de clases utilizando la terminología y notación estándar de la POO. - Desarrollar diagramas de secuencia y colaboración utilizando principios de orientación a objetos. - Evaluar los costos de desarrollo de software en proyectos orientados a objetos comparados con proyectos estructurados. - Diseñar interfaces y clases abstractas para un sistema orientado a objetos, utilizando terminología y técnicas de POO.	- Precisión. - Orden. - Creatividad. - Organización. - Trabajo en equipo. - Responsabilidad. - Comunicación efectiva. - Orientación a la calidad. - Atención al detalle. - Pensamiento lógico. - Resolución de problema. - Gestión del tiempo. - Confidencialidad. - Seguridad. - Adhesión a las normas.	El participante: - Define los conceptos fundamentales de la Programación Orientada a Objetos (POO), según las definiciones estándares de POO. - Elabora diagramas de clases utilizando la terminología y notación estándar de la POO, según modelado UML (Lenguaje Unificado de Modelado). - Desarrolla diagramas de secuencia y colaboración utilizando principios de orientación a objetos, según los estándares de diseño UML - Diseña interfaces y clases abstractas para un sistema orientado a objetos, utilizando terminología y técnicas de POO, según las mejores prácticas de diseño orientado a objetos - Evidencia creatividad y responsabilidad mientras maneja terminología orientada a objetos, de acuerdo a los estándares.	

RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)	
2. Gestiona el modelado de software y el proceso de desarrollo, para crear diagramas de casos.	- El Proceso OOSD (Desarrollo de Software Orientado a Objetos): • Metodología Software. • Listado de los flujos de trabajo del Proceso de OOSD. • Los roles de trabajo del equipo de Software. • La Recopilación de requisitos de flujo de trabajo. - Flujo de trabajo de propósito y de empleo de Roles: • Actividades de flujo de trabajo y Artefactos. • El flujo de trabajo de análisis de requisitos. • Flujo de trabajo de propósito y de empleo Roles. • La arquitectura de flujo de trabajo. - Las pruebas y los flujos de trabajo de implementación. - Los beneficios de software de modelado. - Concepto de modelo - El por qué el software de modelo - OOSD como modelo Transformaciones. - Definición de UML. - Elementos comunes de UML y Conectores - UML (que es y no es) - Justificación de la necesidad de un Diagrama de Casos de Uso. - Elementos de un diagrama de casos de uso: • Actores. • Casos de uso. • Límite de sistema. • Utilice Asociaciones de casos. • Creación inicial Diagrama de Casos de uso. • La identificación de los casos de uso adicionales. • Caso de uso Elaboración. • El análisis de los patrones de herencia. • Análisis de casos de uso. • Dependencias. - Packaging los casos de uso Vistas.	- Desarrollar diagramas de casos de uso utilizando UML. - Configurar y utilizar herramientas de modelado para definir los elementos y conectores comunes en UML. - Evaluar los diagramas de casos de uso para reflejar adecuadamente los requisitos del sistema y las interacciones entre actores y casos de uso. - Ajustar los diagramas de casos de uso para reflejar adecuadamente los requisitos del sistema y las interacciones entre actores y casos de uso. - Diseñar la arquitectura del flujo de trabajo para análisis de requisitos y desarrollo de casos de uso. - Implementar pruebas de software y flujos de trabajo de implementación.	- Precisión. - Orden. - Creatividad. - Organización. - Trabajo en equipo. - Responsabilidad. - Comunicación efectiva. - Orientación a la calidad. - Atención al detalle. - Pensamiento lógico. - Resolución de problema. - Gestión del tiempo. - Confidencialidad. - Seguridad. - Adhesión a las normas.	El participante: - Define concepto de modelado, según procedimientos. - Desarrolla diagramas de casos de uso utilizando UML, según las normas del modelado UML. - Evalúa y ajusta los diagramas de casos de uso para reflejar adecuadamente los requisitos del sistema y las interacciones entre actores y casos de uso, según los principios de modelado de software. - Diseña la arquitectura del flujo de trabajo para análisis de requisitos y desarrollo de casos de uso, según los estándares de diseño de software. - Evidencia responsabilidad y adhesión a las normas mientras gestiona el modelado de software y el proceso de desarrollo, según los estándares establecidos.

RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)	
3. Crea formularios de casos de uso con una planilla, para crear diagramas de actividades.	- Creación de escenarios y Formularios de casos de uso. - Grabación de escenarios de casos de uso. - Selección de escenarios de casos de uso. - Casos Prácticos. - Creación de formulario de Casos prácticos. - Escritura de un escenario de caso de uso. - Especificaciones suplementarias. - Requisitos no funcionales (NFR). - Glosario de términos. - Casos prácticos. - Describiendo un caso de uso con un diagrama de actividad. - Identificación de los elementos de un diagrama de actividad. - Creación de un diagrama de actividades para un caso de uso.	- Desarrollar formularios de casos de uso utilizando una planilla, para facilitar la creación de diagramas de actividades. - Crear y ajustar un diagrama de actividades para representar visualmente un caso de uso. - Seleccionar escenarios de casos de uso, y elabora formularios prácticos para su análisis y desarrollo. - Grabar escenarios de casos de uso, y elabora formularios prácticos para su análisis y desarrollo. - Utilizar herramientas para la grabación y análisis de escenarios de casos de uso, aplicando técnicas de documentación y modelado. - Desarrollar un glosario de términos y especificaciones suplementarias para complementar la creación y comprensión de los casos de uso.	- Precisión. - Orden. - Creatividad. - Organización. - Trabajo en equipo. - Responsabilidad. - Comunicación efectiva. - Orientación a la calidad. - Atención al detalle. - Pensamiento lógico. - Resolución de problema. - Gestión del tiempo. - Confidencialidad. - Seguridad. - Adhesión a las normas.	El participante: - Identifica los diferentes elementos necesarios en un diagrama de actividad, según las técnicas de modelado UML. - Selecciona y graba escenarios de casos de uso, y elabora formularios prácticos para su análisis y desarrollo, según análisis de requisitos. - Utiliza herramientas para la grabación y análisis de escenarios de casos de uso, aplicando técnicas de documentación y modelado, según los estándares de análisis y documentación de software. - Desarrolla un glosario de términos y especificaciones suplementarias para complementar la creación y comprensión de los casos de uso, según los requisitos del proyecto. - Evidencia creatividad y atención al detalle mientras crea formularios de casos de uso con una planilla, de acuerdo a los estándares exigidos.

CONTINUACIÓN...4

RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)	
4. Maneja las abstracciones claves, para identificar casos relevantes.	- La introducción de Abstracciones claves. - Identificación de posibles abstracciones clave. - Identificación de las abstracciones candidatos. - Candidato Key. - Formulario. - Abstracciones. - Glosario del Proyecto. - Descubriendo Clave. - Uso del análisis CRC. - Selección de una Abstracción Candidato Clave. - La identificación de un caso de uso Relevante. - Responsabilidades Determinación y Colaboradores. - La documentación de una Abstracción clave. - Uso de una tarjeta CRC. - Actualización de la clave candidata.	- Utilizar tarjetas CRC para documentar y evaluar las abstracciones clave. - Utilizar el análisis CRC (Clases, Responsabilidades, Colaboradores) para seleccionar una abstracción candidata clave. - Elabora formularios detallados para capturar y definir responsabilidades y colaboradores de las abstracciones clave. - Aplicar técnicas de identificación de abstracciones claves en la selección de los candidatos más relevantes para el desarrollo de casos de uso. - Actualizar y ajustar las abstracciones candidatas clave.	- Precisión. - Orden. - Creatividad. - Organización. - Trabajo en equipo. - Responsabilidad. - Comunicación efectiva. - Orientación a la calidad. - Atención al detalle. - Pensamiento lógico. - Resolución de problema. - Gestión del tiempo. - Confidencialidad. - Seguridad. - Adhesión a las normas. - Coherencia.	El participante: - Identifica las posibles abstracciones clave utilizando herramientas y técnicas adecuadas, según los principios de análisis de abstracciones. - Utiliza el análisis CRC (Clases, Responsabilidades, Colaboradores) para seleccionar una abstracción candidata clave, según las metodologías de diseño de software. - Aplica técnicas de identificación de abstracciones claves en la selección de los candidatos más relevantes para el desarrollo de casos de uso, según los principios de diseño orientado a objetos. - Actualiza y ajusta las abstracciones candidatas clave, según los cambios en los requisitos del proyecto y el análisis continuo. - Evidencia coherencia y responsabilidad mientras maneja las abstracciones claves, de acuerdo a los estándares exigidos.

CONTINUACIÓN...4

RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)	
5. Construye modelo de ámbito o dominio del problema, para validar el tipo de dominio de una organización.	- Presentación del modelo de dominio. - Identificación de los elementos de un diagrama de clases: •Nodos Clase. •Clase de nodo compartimentos. •Asociaciones. - Creación de un modelo de dominio: •Paso 1 - Dibuja los nodos de clase. •Paso 2 - Dibuje las Asociaciones. •Paso 3 - Etiquetar la Asociación y el papel Nombres. •Paso 4 - Etiquetar la multiplicidad Asociación. • Paso 5 - Dibujar las flechas de navegación. •Validación del Modelo de Dominio. •Identificación de los elementos de un diagrama de objeto: •Nodos de objetos. •Enlaces. •Validación del Modelo de dominio mediante Diagramas de objetos. - Creación de un escenario de diagrama de objeto. - Comparación Diagramas de objetos para validar el modelo de dominio.	- Dibujar flechas de navegación en el modelo de dominio para indicar las relaciones y direcciones entre clases. - Crear un modelo de dominio dibujando nodos de clase, estableciendo asociaciones, etiquetando las asociaciones, nombrando roles y definiendo multiplicidades - Desarrollar un diagrama de objetos para representar instancias del modelo de dominio, incluyendo nodos de objetos y enlaces. - Crear escenarios de diagramas de objetos para diferentes situaciones del dominio del problema, según los requisitos del proyecto y las técnicas de modelado UML. - Realizar la validación del modelo de dominio utilizando diagramas de objetos para verificar la corrección y coherencia del modelo de clases.	- Precisión. - Orden. - Creatividad. - Organización. - Trabajo en equipo. - Responsabilidad. - Comunicación efectiva. - Orientación a la calidad. - Atención al detalle. - Pensamiento lógico. - Resolución de problema. - Gestión del tiempo. - Confidencialidad. - Seguridad. - Adhesión a las normas. - Coherencia.	El participante: - Identifica los elementos esenciales de un diagrama de clases, según los principios de modelado UML. - Dibuja flechas de navegación en el modelo de dominio para indicar las relaciones y direcciones entre clases, según las normas de diagramación UML. - Crea un modelo de dominio dibujando nodos de clase, estableciendo asociaciones, etiquetando las asociaciones, nombrando roles y definiendo multiplicidades, según los pasos establecidos para la construcción de modelos de dominio. - Realiza la validación del modelo de dominio utilizando diagramas de objetos para verificar la corrección y coherencia del modelo de clases, según las prácticas de verificación de modelos de software. - Evidencia organización y resolución de problema mientras construye modelo de ámbito o dominio del problema, según las técnicas de modelado de software establecidas.

RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)	
6. Construye la solución de software, para refinar el diseño de clase y realizar prueba.	- Atributos de Refinación del modelo de dominio. - Perfeccionamiento de los Metadatos (Atributos). - La elección de un lugar apropiado para el tipo de datos. - Creación de atributos derivados. - Encapsulación y su aplicación. - Relaciones Refino Clase. - Tipos de relación. - Navegación. - Asociaciones calificadas. - Métodos de relación. - Resolución de muchos-a-muchas relaciones. - Resolución Asociación de Clases. - Métodos de Refinación. - Anotación Método Comportamiento. - Declaración de Constructores. - Revisión del acoplamiento y la coherencia del modelo. - Revisión de Cohesión. - Creación de componentes con interfaces. - Definición de una estructura de paquetes para la Solución. - Uso de los paquetes UML. - Aplicación de los principios del embalaje. - Aislamiento de los subsistemas y las Estructuras. - El desarrollo de una estructura del paquete para el modelo de solución.	- Aplicar principios de encapsulación y selección del tipo de datos. - Elaborar atributos derivados y perfecciona los metadatos del modelo de dominio. - Desarrollar métodos de relación muchos-a-muchas en el modelo de dominio. - Ajustar el acoplamiento y la cohesión del modelo de dominio, asegurando la coherencia del diseño de clases. - Crear una estructura de paquetes para la solución de software, aplicando los principios del embalaje UML.	- Precisión. - Orden. - Creatividad. - Organización. - Trabajo en equipo. - Responsabilidad. - Comunicación efectiva. - Orientación a la calidad. - Atención al detalle. - Pensamiento lógico. - Resolución de problema. - Gestión del tiempo. - Confidencialidad. - Seguridad. - Adhesión a las normas. - Coherencia.	El participante: - Identifica diferentes atributos del modelo de dominio , según procesos técnicos. - Aplica principios de encapsulación y selección del tipo de datos, según los principios de diseño orientado a objetos. - Elabora atributos derivados y perfecciona los metadatos del modelo de dominio, según los principios de diseño orientado a objetos. - Desarrolla métodos de relación muchos-a-muchas en el modelo de dominio, según las técnicas de refinación de diseño. - Evidencia trabajo en equipo y disciplina mientras construye la solución de software, de acuerdo a los procesos técnicos.

CONTINUACIÓN...4

RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)	
	- Mapeo del modelo de dominio de la tecnología Java. - Código de clase. - Type Información. - Atributos. - Asociaciones. - Definición de Sistema de prueba. - Pruebas unitarias. - Las pruebas de integración. - Pruebas Funcionales. - El desarrollo de una prueba de funcionamiento. - Funcionales de Entradas del ensayo. - Identificación de Resultados de la Prueba Funcional. - Identificación de condiciones de prueba funcional. - Creación Funcional. - ebas variantes. - La documentación de casos de prueba.			

Estrategias de enseñanza y aprendizaje

Las estrategias didácticas deberán centrarse en las personas participantes, favoreciendo que este gestione y construya activamente su aprendizaje. La misma deberá propiciar en el participante el trabajo colaborativo, cierto nivel de autonomía y compromiso, el pensamiento crítico, la investigación, convivencia y resolución de problemas ligeramente complejos.

De igual manera, cada una de las estrategias utilizadas deberá permitir el empoderamiento de las personas participantes, donde desarrollen la capacidad de investigar, emitir opinión y tomar decisión con relación a la forma de solucionar los problemas planteados, según su mundo social y laboral. Todo esto implica la construcción de habilidades blandas, además de las de carácter técnico, y que este debe ir en correspondencia con lo que espera el sector productivo y la sociedad.

Las estrategias y actividades de enseñanza y aprendizaje recomendadas para contenidos de las áreas cognoscitiva y actitudinal.

Investigo- documento y aprendo, peceras, debates activos, mapas mentales, comunidades de aprendizaje, Phillip 66, redes de palabras, aprendizaje basado en problema, el aprendizaje basado en proyectos y otras que puedan contribuir al logro del aprendizaje.

Las estrategias y actividades de enseñanza y aprendizaje recomendadas para contenidos del área psicomotora.

Demostración / ejecución, construyendo- aprendo, simulaciones y otras que puedan contribuir al logro del aprendizaje.

De igual forma, el Facilitador podrá utilizar todas las estrategias que considere necesarias para asegurar el logro de las competencias según el programa.

Todas las estrategias deben ser congruentes y pertinentes con el contenido y los resultados de aprendizaje y deberán incluir:

Actividades individuales, en equipo, en línea y en el contexto real de trabajo, entre otros.

Nota: Durante el desarrollo o impartición de los módulos que comprenden un programa de formación, el docente tendrá la autonomía de impartir contenidos necesarios para lograr completar los resultados de aprendizaje que sean innovadores según la tendencia, actualidad, relacionados con la acción formativa que imparte.

El facilitador tendrá que dejar evidencias de la impartición de estos contenidos, registrando la temática impartida, en el programa de formación en el punto destinado a Sugerencias de Mejora para la Revisión del Programa de Formación.

MÓDULO DE APRENDIZAJE No. 5

Código del Módulo	DENOMINACIÓN DEL MÓDULO DE APRENDIZAJE				
	Introducción a la base de datos				
Correspondencia con la unidad de competencia:	UCE_INCO_0005_TEC:				
Competencia final del módulo:	Al finalizar el módulo de aprendizaje, las personas participantes estarán en la capacidad de diseñar y manejar bases de datos, bajo supervisión con responsabilidad sobre el trabajo propio.				
DURACIÓN EN HORAS:	55 horas.	HORAS TEÓRICAS:	16 horas.	HORAS PRÁCTICAS:	39 horas.
RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN	
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)		
1. Diseña sistema de base de datos, para ser implementado y ajustado.	- Concepto de datos. - Concepto de base de datos. - Enfoque de bases de datos. - Clases de usuarios. - Actores (Roles) en un escenario de Base de Datos. - Ventajas de un SGBD o Motor de Base de Datos. - Diseño conceptual de la base de datos: - Ciclo de vida de un Sistema de Información. - Ciclo de Vida del Sistema de Aplicación de Base de Datos. - El proceso de diseño de Base de Datos. - Capacidades de la sentencia SELECT. - Ejecución de la sentencia SELECT. - Recopilación de información con los Usuarios del Negocio. - Técnicas de especificación de requisitos. - La necesidad de modelar los datos. - Modelado Entidad Relación ER. - Tipos de Entidades, Tipos de Relaciones, Atributos Clave, restricciones de cardinalidad. - Identificación de las entidades, atributos y relaciones en un documento de requerimiento de negocios. - Costo de Adquisición del software. - Costo de Mantenimiento. - Costos de Creación y Conversión de la Base de Datos. - Costo de Personal. - Costo de Entrenamiento y de operación.	- Recopilar información detallada de los usuarios del negocio, aplicando técnicas de especificación de requisitos, según los principios de diseño de bases de datos. - Realizar esquema del ciclo básico de procesamiento de información. - Modelar la estructura de datos utilizando diagramas de Entidad-Relación (ER), según las especificaciones del documento de requerimiento. - Desarrollar el diseño conceptual de la base de datos, según procedimientos. - Aplicar la sentencia SELECT y sus capacidades para la gestión y recuperación de datos, según los requisitos de los usuarios. - Optimizar el proceso de diseño de la base de datos, según las necesidades del sistema. - Desarrollar un plan de implementación y ajuste continuo del sistema de base de datos, según procedimientos.	- Precisión. - Orden. - Creatividad. - Organización. - Trabajo en equipo. - Responsabilidad. - Comunicación efectiva. - Orientación a la calidad y al detalle. - Pensamiento lógico. - Resolución de problema. - Gestión del tiempo. - Confidencialidad. - Seguridad. - Adhesión a las normas.	El participante: - Identifica los diferentes tipos de usuarios y actores (roles) en un escenario de base de datos, según el diseño del sistema. - Desarrolla el diseño conceptual de la base de datos, según procedimientos. - Optimiza el proceso de diseño de la base de datos, según las necesidades del sistema. - Desarrolla un plan de implementación y ajuste continuo del sistema de base de datos, según procedimientos. - Demuestra responsabilidad e iniciativa mientras diseña sistema de base de datos, según los requerimientos exigidos.	

RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)	
2. Diseña un esquema relacional, para la creación de tablas en una base de datos utilizando aplicaciones SQL.	- Algoritmo de Transformación de ER en Relacional - Paso 1: Regla aplicada a las Entidades. - Paso 2: Regla aplicada a Entidades Débiles. - Paso 3: Regla aplicada a las relaciones 1:1. - Paso 4: Regla aplicada a las relaciones 1:N. - Paso 5: Regla Aplicada a las Relaciones M:N. - Paso 6: Regla aplicada a los atributos Multivaluados. - Vistas simples y complejas. - Recuperación de datos de vistas. - Creación y mantenimiento de los índices.	- Aplicar la regla de transformación para convertir entidades y entidades débiles del modelo ER a tablas relacionales, según los pasos del algoritmo de transformación. - Crear vistas simples y complejas en SQL para la recuperación eficiente de datos, según los requisitos de las consultas. - Implementar la transformación de relaciones (1:1, 1, M) del modelo ER a un esquema relacional, según las técnicas de normalización establecidas. - Desarrollar índices en SQL para optimizar el acceso y la recuperación de datos en tablas.	- Precisión. - Orden. - Creatividad. - Organización. - Trabajo en equipo. - Responsabilidad. - Comunicación efectiva. - Orientación a la calidad y al detalle. - Pensamiento lógico. - Resolución de problema. - Gestión del tiempo. - Confidencialidad. - Seguridad. - Adhesión a las normas.	El participante: - Identifica las reglas de transformación de un modelo entidad-relación (ER) a un modelo relacional, según los principios establecidos. - Aplica la regla de transformación para convertir entidades y entidades débiles del modelo ER a tablas relacionales, según los pasos del algoritmo de transformación. - Crea vistas simples y complejas en SQL para la recuperación eficiente de datos, según los requisitos de las consultas - Implementa la transformación de relaciones (1:1, 1, M) del modelo ER a un esquema relacional, según las técnicas de normalización establecidas. - Evidencia responsabilidad mientras diseña un esquema relacional, según requerimientos y procesos exigidos.

RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)	
3. Realiza consultas y subconsultas, para aplicar diferentes procesos en SQL.	- Factores que influyen el diseño físico de la base de datos. - Análisis de consultas y Transacciones. - Decisiones de Diseño de los Índices. - Lenguaje SQL, DDL, LDA. - Sentencia SQL para crear una Base de Datos. - Sentencia SQL para crear una Tabla. - Sentencia SQL para crear un Índice. - Sentencia SQL para crear o insertar un dato. - Sentencia SQL para actualizar un dato. - Sentencia SQL para eliminar un dato. - Sentencia SQL para recuperar información. - Sentencia SQL para recuperar información de más de una tabla. - Tipos de funciones disponibles en SQL: caracteres, número y fecha en las instrucciones SELECT. - Límite de las filas recuperadas por una consulta. - Orden de las filas recuperadas por una consulta.	- Implementar sentencias SQL para crear bases de datos y tablas. - Aplicar sentencias SQL para insertar, actualizar y eliminar datos, manteniendo la integridad y consistencia de la base de datos. - Crear consultas SQL utilizando funciones de caracteres, números y fechas, para extraer y manipular datos específicos de la base de datos. - Realizar subconsultas en SQL para recuperar información de múltiples tablas, asegurando la correcta integración de datos entre diferentes fuentes. - Desarrollar índices en SQL para optimizar el rendimiento de las consultas, basándose en el análisis de consultas y transacciones. - Ajustar las consultas SQL para mejorar el rendimiento y eficiencia, mediante el análisis de transacciones y la optimización de sentencias SQL. - Diseñar consultas SQL avanzadas para realizar operaciones complejas en la base de datos, utilizando subconsultas, funciones agregadas, y cláusulas de filtrado. - Ejecutar consultas SQL avanzadas para realizar operaciones complejas en la base de datos, utilizando subconsultas, funciones agregadas, y cláusulas de filtrado.	- Precisión. - Orden. - Creatividad. - Organización. - Trabajo en equipo. - Responsabilidad. - Comunicación efectiva. - Orientación a la calidad y al detalle. - Pensamiento lógico. - Resolución de problema. - Gestión del tiempo. - Confidencialidad. - Seguridad. - Adhesión a las normas.	El participante: - Identifica los diferentes factores que influyen en el diseño físico de una base de datos, según instrucciones establecidas. - Realiza subconsultas en SQL para recuperar información de múltiples tablas, asegurando la correcta integración de datos entre diferentes fuentes, Según las técnicas de consulta SQL. - Desarrolla índices en SQL para optimizar el rendimiento de las consultas, basándose en el análisis de consultas y transacciones, según las decisiones de diseño de índices. - Diseña y ejecuta consultas SQL avanzadas para realizar operaciones complejas en la base de datos, utilizando subconsultas, funciones agregadas, y cláusulas de filtrado, Según las necesidades de procesos específicos en SQL. - Evidencia trabajo en equipo y resolución de problemas mientras realiza consultas y subconsultas, según las necesidades del sistema de gestión de bases de datos.

RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)	
	- Tipos de funciones de conversión que están disponibles en SQL. - Utilización de las funciones de conversión de Datos. - Expresiones condicionales en una sentencia SELECT. - Funciones de grupo disponibles. - Uso de las funciones de grupo. - Agrupación de los datos mediante el uso de la cláusula GROUP BY. - Inclusión y exclusión de filas agrupadas mediante el uso de la cláusula HAVING. - Las sentencias SELECT para datos de acceso de más de una tabla utilizando equijoins y nonequijoins. - Join a una tabla a sí mismo mediante el uso de un self-Join. - Visualización de datos que generalmente no cumple con una condición de unión mediante el uso de los outer Joins. - Generación de un producto cartesiano de todas las filas a partir de dos o más tablas. - Subconsultas. - Tipos de problemas que las subconsultas pueden resolver. - Los tipos de subconsultas. - Ejecución de una sola fila. - Subconsultas de varias filas. - Operadores de conjunto. - Operador de conjunto para combinar varias consultas en una sola consulta. - Uso del control del orden de filas devueltas. - Sentencia del lenguaje de manipulación de datos (DML). - Inserción de filas en una tabla. - Actualización de filas de una tabla. - Eliminación de filas de una tabla. - Control de Transacciones. - Categoría de los principales objetos de la base. - Revisión de la estructura de la tabla. - Los tipos de datos que están disponibles para las columnas. - Creación de una tabla simple. - Las restricciones en el momento de la creación de la tabla. - Función de los objetos de esquema.			

ESTRATEGIAS METODOLÓGICAS PARA EL PLANEAMIENTO DIDÁCTICO

Estrategias de enseñanza y aprendizaje

Las estrategias didácticas deberán centrarse en las personas participantes, favoreciendo que este gestione y construya activamente su aprendizaje. La misma deberá propiciar en el participante el trabajo colaborativo, cierto nivel de autonomía y compromiso, el pensamiento crítico, la investigación, convivencia y resolución de problemas ligeramente complejos.

De igual manera, cada una de las estrategias utilizadas deberá permitir el empoderamiento de las personas participantes, donde desarrollen la capacidad de investigar, emitir opinión y tomar decisión con relación a la forma de solucionar los problemas planteados, según su mundo social y laboral. Todo esto implica la construcción de habilidades blandas, además de las de carácter técnico, y que este debe ir en correspondencia con lo que espera el sector productivo y la sociedad.

Las estrategias y actividades de enseñanza y aprendizaje recomendadas para contenidos de las áreas cognoscitiva y actitudinal.

Investigo- documento y aprendo, peceras, debates activos, mapas mentales, comunidades de aprendizaje, Phillip 66, redes de palabras, aprendizaje basado en problema, el aprendizaje basado en proyectos y otras que puedan contribuir al logro del aprendizaje.

Las estrategias y actividades de enseñanza y aprendizaje recomendadas para contenidos del área psicomotora.

Demostración / ejecución, construyendo- aprendo, simulaciones y otras que puedan contribuir al logro del aprendizaje.

De igual forma, el Facilitador podrá utilizar todas las estrategias que considere necesarias para asegurar el logro de las competencias según el programa.

Todas las estrategias deben ser congruentes y pertinentes con el contenido y los resultados de aprendizaje y deberán incluir:

Actividades individuales, en equipo, en línea y en el contexto real de trabajo, entre otros.

Nota: Durante el desarrollo o impartición de los módulos que comprenden un programa de formación, el docente tendrá la autonomía de impartir contenidos necesarios para lograr completar los resultados de aprendizaje que sean innovadores según la tendencia, actualidad, relacionados con la acción formativa que imparte.

El facilitador tendrá que dejar Demuestras de la impartición de estos contenidos, registrando la temática impartida, en el programa de formación en el punto destinado a Sugerencias de Mejora para la Revisión del Programa de Formación.

MÓDULO DE APRENDIZAJE No. 6

Código del Módulo	DENOMINACIÓN DEL MÓDULO DE APRENDIZAJE				
	Programación en Lenguaje C++				
Correspondencia con la unidad de competencia:	UCE_INCO_0006_TEC:				
Competencia final del módulo:	Al finalizar el módulo, las personas participantes estarán en capacidad de programar en lenguaje C++, bajo supervisión con responsabilidad sobre el trabajo propio.				
DURACIÓN EN HORAS:	50 horas.	HORAS TEÓRICAS:	15 horas.	HORAS PRÁCTICAS:	35 horas.
RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN	
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)		
1. Crea variables y conversión de datos, para almacenar archivos y directorios desde objective C++.	- Conceptos: - Programa. - Tipos de programas. - Código fuente. - Código objeto. - Software de sistema. - Software de aplicación. - Palabras reservadas. - Comentarios. - Concepto de librería. - Concepto de Método o función. - Método principal main. - Conceptos: • Datos. • Variables. • Constantes. - Tipos de datos: • Numéricos: ○ Entero. ○ Real. • Decimal. • Caracteres: ○ Carácter. • Lógicos:	- Declarar y utilizar variables y constantes, según las prácticas de programación orientada a objetos. - Aplicar la conversión de datos entre diferentes tipos en Objective-C++. - Crear y manejar estructuras de control de selección e iteración, incluyendo if simple, if-else, if-else if, switch, for, while, y estructuras anidadas. - Implementar y manejar métodos y funciones en Objective-C++, incluyendo el método principal main, para modularizar el código y definir la funcionalidad principal del programa. - Gestionar archivos y directorios en Objective-C++, realizando operaciones como creación, eliminación, comparación y manejo de permisos, - Desarrollar y utilizar componentes gráficos en Objective-C++.	- Precisión. - Orden. - Creatividad. - Organización. - Trabajo en equipo. - Responsabilidad. - Comunicación efectiva. - Orientación a la calidad. - Atención al detalle. - Pensamiento lógico. - Resolución de problema. - Gestión del tiempo. - Confidencialidad. - Seguridad. - Adhesión a las normas. - Coherencia. -	El participante: - Explica el proceso de creación de variables y la conversión de datos, según procedimientos e instrucciones. - Declara y utiliza variables y constantes, según las prácticas de programación orientada a objetos. - Crea y maneja estructuras de control de selección e iteración, incluyendo if simple, if-else, if-else if, switch, for, while, y estructuras anidadas, según las normas de control de flujo en programación.. - Implementa y maneja métodos y funciones en Objective-C++, incluyendo el método principal main, para modularizar el código y definir la funcionalidad principal del programa, según los principios de diseño de funciones y métodos.. - Evidencia pensamiento lógico y atención al detalle mientras crea variables y conversión de datos, según normativas establecidas.	

RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)	
	○ Boléanos. - Declaraciones: • Variables. • Constantes. - Diccionario. - Índices. - Acceso a elementos de un diccionario. - Eliminación de elementos de un diccionario. - Manejo de directorios. - Creación de directorios. - Atributos de un directorio. - Cambiar directorios. - Borrar directorios. - Comparación de archivos. - Permisos de archivos. - Manejo de archivos. - Componentes gráficos de Objective C++: • Activity. • Intent. • Componentes gráficos para las aplicaciones. - Estructuras de control de selección: • If simple. • If – else. • If – else if. - Case (swicht). - Estructuras de control de iteración: • For. • While. • Estructuras anidadadas.			

RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)	
	- Estructuras de control de selección: • If simple. • If - else. • If - else if. - Case (switch). - Estructuras de control de iteración: • For. • While. - Estructuras anidadas.			
2. Crea arreglos, para insertar y añadir elementos.	- Tipos de arreglos. - Acceso a los elementos de un arreglo. - Borrar elementos de un arreglo. - Conceptos: • Archivo. • Archivo secuencial. • Archivo Indexado. • Acceso secuencial. • Acceso Aleatorio. • Acceso Directo. • Archivo Binario. - Concepto: • Arreglo, array, vector. • Arreglo unidimensional. • Arreglo bidimensional. • Arreglo multidimensional • Índice. - Declaración de un arreglo. - Uso de arreglos.	- Acceder a elementos específicos de un arreglo utilizando índices. - Declarar y utilizar arreglos unidimensionales, bidimensionales y multidimensionales. - Insertar elementos en diferentes tipos de arreglos. - Eliminar elementos de un arreglo. - Utilizar archivos secuenciales y aleatorios en combinación con arreglos. - Implementar el acceso directo a archivos indexados mediante el uso de arreglos. - Gestionar archivos binarios y utilizar arreglos para manipular los datos.	- Precisión. - Orden. - Creatividad. - Organización. - Trabajo en equipo. - Responsabilidad. - Comunicación efectiva. - Orientación a la calidad. - Atención al detalle. - Pensamiento lógico. - Resolución de problema. - Gestión del tiempo. - Confidencialidad. - Seguridad. - Adhesión a las normas. - Coherencia.	El participante: - Explica el proceso de creación e inserción de arreglos, según explicaciones dadas. - Declara y utiliza arreglos unidimensionales, bidimensionales y multidimensionales, según las técnicas de manejo de datos en programación. - Elimina elementos de un arreglo, según las prácticas de gestión dinámica de datos. - Implementa el acceso directo a archivos indexados mediante el uso de arreglos, según los principios de optimización y eficiencia en el manejo de datos. - Demuestra orden y creatividad mientras crea de arreglos, según procedimientos exigidos.

CONTINUACIÓN...6

RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)	
3. Ejecuta diferentes funciones de recursividad y sobrecarga, para realizar declaraciones e interacciones.	- Estructuras de selección. - Recursividad. - ¿Qué es un proceso recursivo? - ¿Dónde se aplica la recursividad? - Verificación de funciones y procedimientos recursivos. - Escritura de programas recursivos. - Ventajas y desventajas. - Características de los problemas que pueden ser resueltos de manera recursiva. - If, if-else. - Switch. - Sobrecarga de métodos. - Recursividad vs Iteración. - Estructuras de iteración. - Declaración de funciones: • For. • While. • Do while. • Labeled. • Continue. • For...in. • For...of. - Anidamientos. - Break y continue. - Funciones. - Parámetros. - Retorno. - Prototipos. • Características. • Procedimiento • Parámetros - Retorno - Tipos de prototipos. - Tipos de funciones:	- Declarar y utilizar estructuras de selección (if, if-else, switch) y estructuras de iteración (for, while, do-while), según las mejores prácticas de programación. - Utilizar prototipos de funciones para declarar y definir funciones con diferentes tipos de parámetros y retornos, según las prácticas de diseño de software. - Aplicar la sobrecarga de métodos y diferencia entre recursividad e iteración, según las necesidades del problema. - Verificar funciones y procedimientos recursivos, según las técnicas de programación recursiva. - Desarrollar y utilizar prototipos de interfaz de usuario, según los principios de diseño de interfaces. - Aplicar técnicas de recursividad y optimización en problemas complejos, según las características del problema.	- Precisión. - Orden. - Creatividad. - Organización. - Trabajo en equipo. - Responsabilidad. - Comunicación efectiva. - Orientación a la calidad. - Atención al detalle. - Pensamiento lógico. - Resolución de problema. - Gestión del tiempo. - Confidencialidad. - Seguridad. - Adhesión a las normas. - Coherencia.	El participante: - Analiza las funciones y la aplicación de recursividad y sobrecarga, según procedimientos. - Declara y utiliza estructuras de selección (if, if-else, switch) y estructuras de iteración (for, while, do-while), según las mejores prácticas de programación. - Utiliza prototipos de funciones para declarar y definir funciones con diferentes tipos de parámetros y retornos, según las prácticas de diseño de software. - Aplica la sobrecarga de métodos y diferencia entre recursividad e iteración, según las necesidades del problema. - Evidencia responsabilidad mientras ejecuta diferentes funciones de recursividad y sobrecarga, según procesos establecidos.

RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)	
	• Funciones que no retornan y no reciben parámetros. • Funciones que retornan y no reciben parámetros. • Funciones que no retornan y reciben parámetros. • Funciones que retornan y reciben parámetros. - Prototipos de interface de usuario. - Ventajas de los prototipos. - Desventajas.			

RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)	
4. Gestiona funciones virtuales, para manejar métodos puros y abstractos.	- Sobrecarga de operadores. - Funciones operador. - Conversiones de tipos. - Operadores y objetos grandes. - Asignación e inicialización. - Subíndices. - Llamadas a función. - De referencia. - Incremento y decremento. - 69 sobrecarga de new y delete. - Funciones amigas o métodos. - 70 templates. - Genericidad. - Funciones genéricas. - Clases genéricas. - Manejo de excepciones. - Programación y errores. - Tratamiento de excepciones en C++ (throw - catch - try). - Entrada y salida. - Introducción. - Objetos stream. - Ficheros.	- Declarar y utilizar operadores y objetos grandes, y aplica asignación e inicialización en C++. - Implementar sobrecarga de operadores y funciones operador en C++. - Utilizar conversiones de tipos y subíndices para mejorar la flexibilidad y precisión del manejo de datos en C++. - Desarrollar y aplicar funciones genéricas y clases genéricas usando templates en C++. - Maneja la sobrecarga de new y delete para gestionar la memoria dinámica en C++. - Aplicar funciones amigas o métodos y gestiona el incremento y decremento en C++. - Desarrollar y gestionar excepciones en C++, aplicando tratamiento de excepciones con throw, catch y try. - Gestionar funciones virtuales, incluyendo métodos puros y abstractos en C++, para permitir el polimorfismo y la implementación de clases abstractas.	- Precisión. - Orden. - Creatividad. - Organización. - Trabajo en equipo. - Responsabilidad. - Comunicación efectiva. - Orientación a la calidad. - Atención al detalle. - Pensamiento lógico. - Resolución de problema. - Gestión del tiempo. - Confidencialidad. - Seguridad. - Adhesión a las normas. - Coherencia.	El participante: - Define concepto de sobrecarga, según instrucciones recibidas. - Declara y utiliza operadores y objetos grandes, y aplica asignación e inicialización en C++, según requerimientos exigidos. - Desarrolla y gestiona excepciones en C++, aplicando tratamiento de excepciones con throw, catch y try, según los procesos técnicos. - Gestiona funciones virtuales, incluyendo métodos puros y abstractos en C++, para permitir el polimorfismo y la implementación de clases abstractas, según requerimientos. - Evidencia gestión de tiempo y adhesión a las normas mientras gestiona funciones virtuales, según requerimientos exigidos.

CONTINUACIÓN...6

RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)	
5. Crea clases, para diseñar bibliotecas.	- Programación en C++. - Documentación. - Diseño e implementación. - Elección de clases. - Interfaces e implementación. - Librerías de clases. - Diseño de librerías. - Clases contenedor. - Clases para aplicaciones. - Clases de interface. - Eficiencia temporal y gestión de memoria. - Estandarización. - Relación C/C++. - No se puede usar en ansi C. - Diferencias entre C y C++.	- Declarar e implementar clases básicas en C++, utilizando principios de diseño de clases. - Diseñar y documentar librerías de clases en C++, aplicando principios de estandarización - Desarrollar e implementar clases contenedor y clases para aplicaciones en C++, optimizando la gestión de memoria. - Crear e implementar interfaces e implementaciones en C++, incluyendo clases de interfaz. - Diseñar librerías de clases en C++, considerando la eficiencia temporal y la gestión de memoria	- Precisión. - Orden. - Creatividad. - Organización. - Trabajo en equipo. - Responsabilidad. - Comunicación efectiva. - Orientación a la calidad. - Atención al detalle. - Pensamiento lógico. - Resolución de problema. - Gestión del tiempo. - Confidencialidad. - Seguridad. - Adhesión a las normas. - Coherencia.	El participante: - Explica el proceso de creación y jerarquías de clases y el diseño de bibliotecas, según instrucciones dadas. - Diseña y documenta librerías de clases en C++, según principios de estandarización. - Desarrolla e implementa clases contenedor y clases para aplicaciones en C++, optimizando la gestión de memoria, según los estándares establecidos. - Crea e implementa interfaces e implementaciones en C++, incluyendo clases de interfaz, según las directrices de arquitectura de software. - Evidencia pensamientos lógico y atención al detalle mientras crea clases, según requerimientos exigidos.

ESTRATEGIAS METODOLÓGICAS PARA EL PLANEAMIENTO DIDÁCTICO

Estrategias de enseñanza y aprendizaje

Las estrategias didácticas deberán centrarse en las personas participantes, favoreciendo que este gestione y construya activamente su aprendizaje. La misma deberá propiciar en el participante el trabajo colaborativo, cierto nivel de autonomía y compromiso, el pensamiento crítico, la investigación, convivencia y resolución de problemas ligeramente complejos.

De igual manera, cada una de las estrategias utilizadas deberá permitir el empoderamiento de las personas participantes, donde desarrollen la capacidad de investigar, emitir opinión y tomar decisión con relación a la forma de solucionar los problemas planteados, según su mundo social y laboral. Todo esto implica la construcción de habilidades blandas, además de las de carácter técnico, y que este debe ir en correspondencia con lo que espera el sector productivo y la sociedad.

Las estrategias y actividades de enseñanza y aprendizaje recomendadas para contenidos de las áreas cognoscitiva y actitudinal para.

Investigo- documento y aprendo, peceras, debates activos, mapas mentales, comunidades de aprendizaje, Phillip 66, redes de palabras, aprendizaje basado en problema, el aprendizaje basado en proyectos y otras que puedan contribuir al logro del aprendizaje.

Las estrategias y actividades de enseñanza y aprendizaje recomendadas para contenidos del área psicomotora.

Demostración / ejecución, construyendo- aprendo, simulaciones y otras que puedan contribuir al logro del aprendizaje.

De igual forma, el Facilitador podrá utilizar todas las estrategias que considere necesarias para asegurar el logro de las competencias según el programa.

Todas las estrategias deben ser congruentes y pertinentes con el contenido y los resultados de aprendizaje y deberán incluir:

Actividades individuales, en equipo, en línea y en el contexto real de trabajo, entre otros.

Nota: Durante el desarrollo o impartición de los módulos que comprenden un programa de formación, el docente tendrá la autonomía de impartir contenidos necesarios para lograr completar los resultados de aprendizaje que sean innovadores según la tendencia, actualidad, relacionados con la acción formativa que imparte.

El facilitador tendrá que dejar evidencias de la impartición de estos contenidos, registrando la temática impartida, en el programa de formación en el punto destinado a Sugerencias de Mejora para la Revisión del Programa de Formación.

MÓDULO DE APRENDIZAJE No. 7

Código del Módulo	DENOMINACIÓN DEL MÓDULO DE APRENDIZAJE				
	Programación C#				
Correspondencia con la unidad de competencia:	UCE_INCO_0004_CAP:				
Competencia final del módulo:	Al finalizar la cualificación la persona participante estará en capacidad de desarrollar aplicaciones de software utilizando el lenguaje de programación C#, aplicando principios de programación orientada a objetos, estructuras de control, manejo de excepciones y acceso a bases de datos, manejándose en un ambiente de confianza y participación.				
DURACIÓN EN HORAS:	60 horas.	HORAS TEÓRICAS:	18 horas.	HORAS PRÁCTICAS:	42 horas.
RESULTADOS DE APRENDIZAJE	CONTENIDOS				CRITERIOS DE EVALUACIÓN
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)		
1. Crea formularios dentro del entorno gráfico, para desarrollar proyecto.	- Concepto de .Net. - Concepto de objeto - Propiedades - Papel que cumplen las propiedades, métodos y eventos - Terminología de .NET - Concepto de namespace - Concepto de assembly - Concepto de proyecto - Tipo de datos - Variables - Constantes - Operadores, estructuras de control y ciclos repetitivos. - Módulos de formulario. - Módulos estándar. - Ventana de código. - Cuadro lista de objetos. - Cuadro lista de eventos. - La barra de división. - Botón de procedimiento. - Botón de modulo completo. - Pantalla modo edición código. - Carácter de continuación de línea. - Comentarios. - Formularios modales y no modales. - Finalización de una aplicación. - Interfaz de múltiples documentos (mdi).	- Ajustar el entorno de desarrollo seleccionando el namespace y organizando los assembly. - Implementar estructuras de control y ciclos repetitivos. - Configurar adecuadamente la finalización de la aplicación. - Diseñar formularios modales y no modales. - Desarrollar módulos estándar y módulos de formulario.	- Orden. - Iniciativa. - Participación. - Responsabilidad - Iniciativa. - Coherencia. - Precisión. - Atención al detalle. - Razonamiento lógico. - Planificación. - Adhesión a las normas. - Seguridad.		La persona participante: - Define concepto de objetos y propiedades de un formulario, según requerimientos de funcionalidad. - Ajusta el entorno de desarrollo seleccionando el namespace y organizando los assembly, según los requisitos del proyecto. - Diseña formularios modales y no modales, según los requerimientos del usuario. - Desarrolla módulos estándar y módulos de formulario, según los principios de modularidad. - Evidencia creatividad y coherencia mientras crea formularios dentro del entorno gráfico.

RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)	
2. Maneja la programación orientada a objetos, para aplicar clases, métodos y propiedades técnicas.	- Concepto de: - POO - Clase - Objetos - Instanciar - Características de las clases - Estructura de una clase - Cuerpo de una Clase - Propiedades y atributos de una clase - Métodos - Recargado de métodos - Evento y/o mensaje - Abstracción - Encapsulamiento - Modularidad - Ocultación o Aislamiento - Polimorfismo - Herencia - Modificadores de Accesos de las clases, métodos y propiedades.	- Diseñar la estructura de las clases organizando el cuerpo de la clase. - Desarrollar clases con propiedades y métodos. - Implementar el encapsulamiento y la ocultación de datos. - Crear nuevas clases aplicando herencia. - Implementar polimorfismo, según las necesidades del proyecto.	- Orden. - Iniciativa. - Participación. - Responsabilidad - Iniciativa. - Coherencia. - Precisión. - Atención al detalle. - Razonamiento lógico. - Planificación. - Adhesión a las normas. - Seguridad.	La persona participante: - Define concepto de POO, según procedimientos. - Desarrolla clases con propiedades y métodos, según los requerimientos de funcionalidad. - Implementa el encapsulamiento y la ocultación de datos, de acuerdo a las propiedades y métodos críticos. - Crea nuevas clases aplicando herencia, según las especificaciones de diseño. - Evidencia pensamiento lógico y adhesión a las normas mientras maneja la programación orientada a objetos, según las especificaciones técnicas.

RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)	
3. Integra y configura controles estándar, avanzados y adicionales, para la gestión de datos y la interacción con el usuario en un proyecto.	- Label - TextBox - Button - Casilla de control de verificación (checkbox) - Control de barra de desplazamiento - Barra de desplazamiento horizontal (hscrollbar) - Barra de desplazamiento vertical (vscrollbar) - Control cuadro de lista (listbox) - Control cuadro combinado (combobox) - Uso del examinador de objetos para ver procedimientos y librerías de objetos. - Control cuadro de lista de unidades (drivelistbox) - Control de cuadro de lista de directorios (dirlistbox) - Control cuadro de lista de archivos (filelistbox) - Control de imagen (image) - Función load picture() - Control temporizador (timer) - Uso de control data - Uso de controles enlazados datos active x. - Conexión a Bases de Datos - Insertar Datos - Consultar datos - Eliminar datos - Editar Datos - Consulta y presentación datos - Uso control dbgrid - Uso control msflexgrid - Uso control dbcombo.	- Añadir controles estándar como Label, TextBox, Button, y Checkbox en el formulario - Ajustar controles estándar como Label, TextBox, Button, y Checkbox en el formulario. - Implementar controles avanzados como ListBox, ComboBox, HScrollBar, VScrollBar, y Image. - Personalizar controles avanzados como ListBox, ComboBox, HScrollBar, VScrollBar, y Image. - Incorporar controles adicionales como DriveListBox, DirListBox, FileListBox, y Timer. - Configurar controles adicionales como DriveListBox, DirListBox, FileListBox, y Timer. - Implementar controles de datos como DBGrid, MSFlexGrid, y DBCombo. - Utilizar controles de datos como DBGrid, MSFlexGrid, y DBCombo.	- Orden. - Iniciativa. - Participación. - Responsabilidad - Iniciativa. - Coherencia. - Precisión. - Atención al detalle. - Razonamiento lógico. - Planificación. - Adhesión a las normas. - Seguridad.	La persona participante: - Define concepto de casilla de control de verificación, según procedimientos. - Añade y ajusta controles estándar como Label, TextBox, Button, y Checkbox en el formulario, según los requisitos del proyecto. - Implementa y personaliza controles avanzados como ListBox, ComboBox, HScrollBar, VScrollBar, y Image, según las necesidades de navegación y visualización de datos. - Implementa y utiliza controles de datos como DBGrid, MSFlexGrid, y DBCombo, según las especificaciones del proyecto. - Evidencia responsabilidad y adhesión las normas mientras integra y configura controles estándar, avanzados y adicionales, según especificaciones establecidas.

RESULTADOS DE APRENDIZAJE	CONTENIDOS			CRITERIOS DE EVALUACIÓN
	SABER (CONOCIMIENTOS)	SABER HACER (HABILIDADES COGNITIVAS Y PRÁCTICAS)	SABER SER (HABILIDADES CONDUCTUALES)	
4. Utiliza herramientas depuración, validación de datos y manipulación de errores, para ejecutar el sistema de depuración, validación de datos y manipulación de errores.	- Herramientas de depuración - Puntos de interrupción y expresiones de interrupción - Expresiones de inspección - Ventana de inmediato - Ventanas locales - Validación de datos - Restricción de opciones con controles - Validación de información a nivel de formulario - Validación de todos los campos de un formulario - Uso del evento key run load - Manipulación de errores en tiempo de ejecución - Entendiendo el proceso de manipulación de errores - Prioridades del objeto error.	- Utilizar el evento KeyDown y Load para validar la información en tiempo real. - Implementar técnicas de validación de datos a nivel de formulario. - Utilizar herramientas de inspección y ventanas de depuración para análisis detallado. - Configurar el proceso de manejo de errores.	- Orden. - Iniciativa. - Participación. - Responsabilidad - Iniciativa. - Coherencia. - Precisión. - Atención al detalle. - Razonamiento lógico. - Planificación. - Adhesión a las normas. - Seguridad.	La persona participante: - Explica el proceso de manipulación de errores, según procedimientos. - Implementa técnicas de validación de datos a nivel de formulario, según los requisitos del proyecto. - Utiliza herramientas de inspección y ventanas de depuración para análisis detallado, según las especificaciones establecidas. - Configura el proceso de manejo de errores, según las necesidades del sistema. - Evidencia responsabilidad y adhesión a las normas mientras utiliza herramientas depuración, validación de datos y manipulación de errores, según procedimientos.

ESTRATEGIAS METODOLÓGICAS PARA EL PLANEAMIENTO DIDÁCTICO

Estrategias de enseñanza y aprendizaje.

Las estrategias didácticas deberán centrarse en las personas participantes, favoreciendo que este gestione y construya activamente su aprendizaje. La misma deberá propiciar en el participante el trabajo colaborativo, compromiso, el pensamiento crítico, comunicación de opiniones, convivencia y resolución de problemas sencillos.

De igual manera, cada una de las estrategias utilizadas deberá permitir el empoderamiento de las personas participantes, donde desarrollen la capacidad de investigar, emitir opinión y tomar decisión con relación a la forma de solucionar los problemas rutinarios planteados, según su mundo social y laboral. Todo esto implica la construcción de habilidades blandas, además de las de carácter técnico, y que este debe ir en correspondencia con lo que espera el sector productivo y la sociedad.

Las estrategias y actividades de enseñanza y aprendizaje recomendadas para contenidos de las áreas cognoscitiva y actitudinal son:

Lupa indagatoria, apuntes colaborativos, preguntas en pares, lecturas reflexivas, enseñanzas orientadas, dramatizaciones, juego de roles y otras que puedan contribuir al logro del aprendizaje y otras que puedan contribuir al logro del aprendizaje.

Las estrategias y actividades de enseñanza y aprendizaje recomendadas para contenidos del área psicomotora son:

Construyendo – aprendo, enseñando aprendo, demostración / ejecución, simulación, prácticas de taller o laboratorio y otras que puedan contribuir al logro del aprendizaje.

De igual forma el Facilitador podrá utilizar todas las estrategias que considere necesarias para asegurar el logro de las competencias según el programa.

Todas las estrategias deben ser congruentes y pertinentes con el contenido y los resultados de aprendizaje y deberán incluir:

Actividades individuales, en equipo, en línea y en el contexto real de trabajo, entre otros.

Nota: Durante el desarrollo o impartición de los módulos que comprenden un programa de formación, el docente tendrá la autonomía de impartir contenidos necesarios para lograr completar los resultados de aprendizaje que sean innovadores según la tendencia, actualidad, relacionados con la acción formativa que imparte.

El facilitador tendrá que dejar evidencias de la impartición de estos contenidos, registrando la temática impartida, en el programa de formación en el punto destinado a Sugerencias de Mejora para la Revisión del Programa de Formación.

6 PERFIL TÉCNICO PEDAGÓGICO DEL FACILITADOR (A) DEL PROGRAMA

Programación básica de computadora

1. Requisitos Mínimos:

1.1) Educación Básica:

- Técnico en Informática o Técnico en Desarrollo de Aplicaciones o Técnico en Redes y Comunicaciones de Datos o Licenciatura en Informática o Ingeniería en Sistemas o áreas afines.

1.2) Educación Especializada:

- Facilitador de la Formación Profesional.

1.3) Otros Estudios y/o Habilidades:

- Tener conocimientos básicos del manejo de Procesador de texto o documentos y del manejo de presentaciones.

1.4) Experiencia de Trabajo:

- Un (1) año en labores del oficio en el área de informática o Programación básica de computadora o áreas afines.

1.5) Experiencia Docente:

- Un (1) año (no imprescindible).

1.6) Cualidades Especiales Personales:

- Buena dicción.
- Facilidad de expresión.
- Capacidad analítica.
- Buena presentación.
- Orientado al puesto.

RECOMENDACIONES:

- Tener conocimientos actualizados en el área de Programación básica de computadora.

7 REQUERIMIENTO DE RECURSOS

1. Ambiente de formación en el área de Programación básica de computadora.

- Aula con espacio pedagógico mínimo requerido de 23 mts², con iluminación y ventilación adecuada.
- Para los Centros Operativos del sistema: Aula con espacio pedagógico mínimo requerido de 32 mts² (23 mts² para los participantes y 9 mts² para espacio del facilitador).
- Taller con espacio mínimo requerido de 28 mts², con iluminación y ventilación adecuada.

2. Materiales.

- Papel de copia 8 ½ x 11.

3. Mobiliario.

- Escritorio para facilitador.
- Silla para facilitador.
- Mesas o sillas o butacas para participantes.
- Pizarra magnética o de formica.

4. Equipos.

- Computadora de mesa o laptop, con software de presentaciones instalado.
- Proyector multimedia.
- Espacio para proyección.
- Juego de bocina para sonido.

8 LISTA MAESTRA.

EQUIPOS			
No.	DESCRIPCIÓN	UNIDAD	CANTIDAD
1-	Computadora con sistema operativo Windows instalado	Unidad	18

9 REFERENCIAS DOCUMENTALES.

- 5., A. (2015). Principios del Desarrollo de Aplicaciones Java. Editora Eni.
- Blanquer, E. B. (2016). Desarrollo de Aplicaciones IOS. Editora RaMa.
- Guerin, B. A. (2015). ASP.NET en C# con Visual Estudio, Diseño y Desarrollo de Aplicaciones Web.
- Montaña, P. R. (2016). Introducción a la programación de computadores.
- Ramírez, F. (2016). Introducción a la programación. 2da Edición.
- Remón, M. Á. (2015). Desarrollo de Aplicaciones Web con PHP y MySQL.
- Torres, M. (2015). Desarrollo de Aplicaciones con Visual C#. Editora Alfa & Omega.
- Vaquero, E. R. (2017). Desarrollo de Aplicaciones Móviles multiplataforma con Ionic desde cero.

WEBGRAFÍA

- Registrada, a. e. (2024). Cursos de informática gratuitos online. Obtenido de Cursos de informática gratuitos online: <https://www.aulaclie.es/index.htm>
- Los mejores recursos para aprender diseño y programación web. (2024). Obtenido de Los mejores recursos para aprender diseño y programación web.: <https://uniwebsidad.com/?from=librosweb>

10 RESPONSABLES DEL DISEÑO.

El presente documento corresponde al programa de **Programación básica de computadora, Certificado de Aptitud profesional**. Fue revisado según PT-ONA-052 en la Dirección de Formación Profesional, a través del Departamento de Desarrollo Curricular.

ORGANIZADO Y/O DISEÑADO POR

: **Mario Valdez De Jesús**
Técnico Departamento Desarrollo Curricular
Edificio Corporativo Institucional.

REVISADO POR

: **Kenia Peña**
Encargada Departamento Desarrollo Curricular
Edificio Corporativo Institucional.

VERIFICADO POR

: **Luis H. Beltré M.**
Director de Formación Profesional
Edificio Corporativo Institucional

Andrea Lina Ferreras
Coordinadora de Calidad de Contenido
Edificio Corporativo Institucional.

ESPECIALISTAS TÉCNICOS

: **Rafael Elías, José Grullón, Francisco Hernández, Pedro Omar Vanderlinder Y Rafael Elías Gonzales.**
Dirección Regional Cibao Norte
Facilitadores
Ramón Abreu, Cristian Espinal,
Dirección Regional Metropolitana
Enc. De taller y facilitador
Joan Paulino, E. Encarnación, Williams de los Santos y Fernelis Mateo
Dirección Regional Sur
facilitadores
Kelvin Abreu y Stephany Rivera Sánchez
Dirección Regional Oriental
Enc. De taller y facilitadora
Claudio Arias
Philippe Made
Instituto Técnico Salesiano (ITESA)
Facilitadores

11 Sugerencias de Mejora para la Revisión del Programa de Formación.

Dirección Regional:	Centro Operativo del Sistema (COS):
Nombre de la Acción Formativa:	Facilitador/a:
Fecha:	

Página	Contenido actual	Propuesta	Justificación de la mejora

Firma del Facilitador/a _____ Supervisor _____

NOTA: Si su propuesta toma más espacio del que contiene el formulario, puede anexar hojas adicionales.

Diirección Regional o Nombre del COS: Se escribe el nombre de la Gerencia Regional o COS responsable de la ejecución de la acción formativa.

Acción Formativa: Se escribe el nombre de la acción formativa tal como aparece en la programación.

Facilitador/a: Se escribe el nombre del docente responsable de la acción formativa.

Fecha: Se escribe la fecha real en que se realiza la propuesta de mejora.

Página: Se escribe el número de la página del programa, donde se encuentra la mejora.

Contenido actual: Se escribe la información que se sugiere modificar, tal como aparece en el programa de acción formativa.

Propuesta: Se escribe la sugerencia o propuesta de cambio, de cómo debe aparecer el contenido dentro del programa de formación, se especifica si se debe eliminar, cambiar de orden, fusionar o insertar una nueva información.

Justificación de la mejora: Se escribe por qué existe la necesidad de un cambio el contenido del programa de formación.

Facilitador/a: Se escribe el nombre completo del (a) Facilitador (a) que imparte la acción formativa.

Supervisor: Se escribe el nombre completo del Encargado del Taller/ Asesor (a) de Formación, responsable del programa de formación.

12 GUÍA DE PRÁCTICAS POR MÓDULOS.

Nombre del Programa: Programación básica de computadora Código: 30936

No. del Módulo	Nombre del Módulo	Prácticas a Realizar	Evidencia del Producto Resultante
1.	Introducción al computador y sistema Operativo	1. Configurar las motherboards, procesadores y módulos de memoria (RAM y ROM).	- Las motherboards, procesadores y módulos de memoria (RAM y ROM), configuradas.
		2. Desarrollar y aplicar procedimientos de soporte técnico post-instalación para asegurar la correcta operación del sistema operativo.	- Procedimientos de soporte técnico post-instalación para asegurar la correcta operación del sistema operativo, desarrollados y aplicados.
		3. Desarrollar y ejecutar un plan de optimización del sistema operativo, utilizando herramientas del sistema y software de terceros para mejorar el rendimiento general del equipo.	- Un plan de optimización del sistema operativo, utilizando herramientas del sistema y software de terceros para mejorar el rendimiento general del equipo, desarrollado y ejecutado.
		4. Crear, eliminar, y organizar archivos y carpetas en el Explorador de Windows y en la línea de comandos (cmd) de Windows.	- Archivos y carpetas en el Explorador de Windows y en la línea de comandos (cmd) de Windows, creados, eliminados y organizados.
2.	Introducción a la programación.	1. Codificar estructuras de control condicionales y ciclos repetitivos	- Estructuras de control condicionales y ciclos repetitivos, codificadas.
		2. Desarrollar algoritmos que utilicen vectores y matrices para almacenar y manipular datos.	- Algoritmos que utilicen vectores y matrices para almacenar y manipular datos, desarrollados.
		3. Evaluar y ajustar el sistema de base de datos durante su ciclo de vida, incluyendo la revisión periódica del diseño y la aplicación de mejoras basadas en el feedback de los usuarios y el rendimiento del sistema.	- El sistema de base de datos durante su ciclo de vida, incluyendo la revisión periódica del diseño y la aplicación de mejoras basadas en el feedback de los usuarios y el rendimiento del sistema, evaluado y ejecutado.
3.	Análisis y Diseño de Sistemas I.	1. Diseñar interfaces y clases abstractas para un sistema orientado a objetos, utilizando terminología y técnicas de POO.	- Interfaces y clases abstractas para un sistema orientado a objetos, utilizando terminología y técnicas de POO, diseñadas.
		2. Diseñar la arquitectura del flujo de trabajo para análisis de requisitos y desarrollo de casos de uso.	- La arquitectura del flujo de trabajo para análisis de requisitos y desarrollo de casos de uso, diseñada.
		3. Utilizar herramientas para la grabación y análisis de escenarios de casos de uso, aplicando técnicas de documentación y modelado.	- Herramientas para la grabación y análisis de escenarios de casos de uso, aplicando técnicas de documentación y modelado, utilizadas.
		4. Actualizar y ajustar las abstracciones candidatas clave.	- Las abstracciones candidatas clave, actualizadas y ajustadas.
		5. Crear un modelo de dominio dibujando nodos de clase.	- Un modelo de dominio dibujando nodos de clase, creado.
		6. Desarrollar métodos de relación muchos-a-muchas en el modelo de dominio.	- Métodos de relación muchos-a-muchas en el modelo de dominio, desarrollados.
4.	Introducción a Base de Datos.	1. Desarrollar un plan de implementación y ajuste continuo del sistema de base de datos.	- Un plan de implementación y ajuste continuo del sistema de base de datos, desarrollado
		2. Implementar la transformación de relaciones (1:1, 1, M) del modelo ER a un esquema relacional.	- La transformación de relaciones (1:1, 1, M) del modelo ER a un esquema relacional, implementada.
		3. Ejecutar consultas SQL avanzadas para realizar operaciones complejas en la base de datos, utilizando subconsultas, funciones agregadas, y cláusulas de filtrado.	- Consultas SQL avanzadas para realizar operaciones complejas en la base de datos, utilizando subconsultas, funciones agregadas, y cláusulas de filtrado, ejecutadas.

No. del Módulo	Nombre del Módulo	Prácticas a Realizar	Evidencia del Producto Resultante
5.	Programación en Lenguaje C++.	1. Crear y manejar estructuras de control de selección e iteración, incluyendo if simple, if-else, if-else if, switch, for, while, y estructuras anidadas.	- Estructuras de control de selección e iteración, incluyendo if simple, if-else, if-else if, switch, for, while, y estructuras anidadas, creadas y manejadas.
		2. Implementar el acceso directo a archivos indexados mediante el uso de arreglos.	- El acceso directo a archivos indexados mediante el uso de arreglos, implantado.
		3. Utilizar prototipos de funciones para declarar y definir funciones con diferentes tipos de parámetros y retornos.	- Prototipos de funciones para declarar y definir funciones con diferentes tipos de parámetros y retornos, utilizados.
		4. Desarrollar y gestionar excepciones en C++, aplicando tratamiento de excepciones con throw, catch y try.	- Excepciones en C++, aplicando tratamiento de excepciones con throw, catch y try, desarrolladas y gestionadas.
		5. Crear e implementar interfaces e implementaciones en C++, incluyendo clases de interfaz.	- Interfaces e implementaciones en C++, incluyendo clases de interfaz, creadas e implementadas.
6.	Programación C#.	1. Desarrollar módulos estándar y módulos de formulario.	- Módulos estándar y módulos de formulario, desarrollados.
		2. Crear nuevas clases aplicando herencia.	- Nuevas clases aplicando herencia, creadas.
		3. Implementar y utilizar controles de datos como DBGrid, MSFlexGrid, y DBCombo	- Controles de datos como DBGrid, MSFlexGrid, y DBCombo, implementados y utilizados.
		4. Configurar el proceso de manejo de errores	- El proceso de manejo de errores, configurado.

Nota: en la guía de prácticas por módulos solo están especificadas las prácticas que se tienen que realizar en los módulos técnicos, las prácticas de los módulos básicos y transversales están contenidas en el desarrollo del saber hacer de estas unidades.