1. IDENTIFICACIÓN DE LA GUIA DE APRENIZAJE

- Denominación del Programa de Formación: **PROGRAMACION DE SOFTWARE**
- Código del Programa de Formación: **228120 Vr: 102**
- Nombre del Proyecto (si es formación Titulada): **SOFTWARE PARA CONTROL DEL INVENTARIO DE EQUIPOS INFORMATICOS EN EL INSTITUTO DE ESTUDIOS TECNICOS FORMAR.**
- Fase del Proyecto (si es formación Titulada): **ANALISIS**
- Actividad de Proyecto(si es formación Titulada): **ANALIZAR EL SISTEMA DE INFORMACIÓN SEGÚN DISEÑO ENTREGADO**
- Competencia: **220501032 -ANALIZAR LOS REQUISITOS DEL CLIENTE PARA CONSTRUIR EL SISTEMA DE INFORMACION. = 380 HORAS**
- Resultados de Aprendizaje Alcanzar: **437634 - IDENTIFICAR CADA UNO DE LOS CONCEPTOS Y PRINCIPIOS QUE CONSTITUYE LA PROGRAMACIÓN ORIENTADA A OBJETOS PARA INTREPRETAR EL DISEÑO.**
- Duración de la Guía: **125 horas.**

2. PRESENTACION

- Motivación hacia la actividad de aprendizaje:

Lectura Introductoria: La gente que se dedica a la programación y al desarrollo de software tiene opiniones de lo más diversas sobre lo que caracteriza a la POO. A continuación, mostramos el resultado de una encuesta realizada en marzo y abril de 2015 sobre 250 respuestas obtenidas. A modo de caracterización del universo consultado, informamos:

- El 100% trabaja en desarrollo de software, aunque el 5% tiene menos de 1 año de experiencia. Un 65% dice tener más de 5 años de experiencia y un 28% más de 10 años.
- En cuanto a edades, el grupo etario más numeroso (se dividió en grupos cada 10 años, con edades terminadas en 5) es el de 26 a 35 años, con el 67% del total. El grupo de 25 años o menos sumó un 19% y de 36 a 45, el 12%. De allí se deduce que muy poca gente de más de 46 años completó la encuesta.

Respecto de los lenguajes de programación que dicen utilizar, descartando los que suman menos del 10%, obtuvimos:

- SQL: 73%
- Java: 70%
- JavaScript: 55%
- C++: 30%
- C#: 30%
- PHP: 28%
- C: 27%
- Python: 24%

- Ruby: 10%

Uno de los lenguajes elegidos para este libro, Smalltalk, sólo es conocido por el 6% de los encuestados. Ahora bien, los resultados que nos interesa destacar de la encuesta son los conceptos que los encuestados consideraron "fundamentales" o "muy importantes" dentro de la programación orientada a objetos. De ellos, los 10 más elegidos, fueron, con sus porcentajes:

- Clase: 91%
- Objeto: 90%
- Encapsulamiento: 90%
- Polimorfismo: 88%
- Herencia: 87%
- Interfaces: 76%
- Clases abstractas: 71%
- Ocultamiento de implementación: 71%
- Métodos abstractos: 70%
- Sobrecarga de métodos: 65%

Para los autores, lo que caracteriza a la POO es que un programa es un conjunto de objetos enviándose mensajes y reaccionado ante los mismos, más la idea de que es responsabilidad de cada objeto saber cómo responder a esos mensajes, además del hecho de que cada objeto podría responder a los mensajes de manera distinta.

Por lo tanto, los conceptos centrales de POO para nosotros son: objetos, mensajes, encapsulamiento y polimorfismo. Luego hay conceptos muy interesantes, pero que no hacen a la orientación a objetos en sí misma, sino que son cuestiones de implementación o usos adquiridos de otros paradigmas. Curiosamente, el concepto de mensaje sólo es "fundamental" o "muy importante" para el 48% de los encuestados, mientras que el concepto más importante, con el 91% de los encuestados considerándolo "fundamental" o "muy importante" es el de clase, que es una cuestión de implementación solamente. Otra cosa que llama poderosamente la atención es la cantidad de conceptos que para nuestros encuestados son centrales en la orientación a objetos: hay casi dos tercios que creen que hay 10 conceptos centrales, lo cual parece mucho para cualquier paradigma.

- Guía y organización del aprendizaje: El docente formador conformará grupos de cuatro alumnos para la ejecución de las actividades de trabajo colaborativo, teniendo en cuenta que en cada grupo se encuentre personas de diversos estilos de aprendizaje (Asimilador, Convergente, Divergente y Acomodador).

- Motivación a las acciones: Cada grupo de aprendices escogerá un representante para hacer las veces de monitor de grupo y determinara roles a cada integrante para el trabajo autónomo sistemático y organizado en las actividades a desarrollar.

- Relación de conocimientos previos con los nuevos: Observación y análisis de video interactivo y Lectura y análisis del glosario de términos relacionado.

- Promoción del aprendizaje colaborativo y el crecimiento integral del grupo:

Cada grupo deberá escoger un representante para socializar en plenaria de curso los trabajos realizados.

3. FORMULACION DE LAS ACTIVIDADES DE APRENDIZAJE

- Descripción de la(s) Actividad(es):

Actividad de Enseñanza: Aprendizaje; Evaluación:

Comprender conceptos, lógica y técnicas que constituyen la programación orientada a objetos para la interpretación del diseño de un proyecto. Tiempo máximo: 180 horas.

1° Proceso Pedagógico: Actividad de Reflexión inicial.

1° Momento.- Los alumnos conformados en grupos de 4 personar realizarán la lectura Introductoria de la presente guía de aprendizaje.

2° Momento.- Los alumnos conformados en grupos de 4 personar realizarán la lectura del glosario de la presente guía de aprendizaje.

3° Momento.- Cada grupo escogerá un representante para socializar en plenaria de curso los conocimientos adquiridos en los momentos 1 y 2.

2° Proceso Pedagógico: Contextualización e identificación de conocimientos:

4° Momento.- Los aprendices conformados en grupos de cuatro personas desarrollarán el siguiente investigación: FUNDAMENTOS DE LA LÓGICA DE PROGRAMACIÓN, mediante el uso de las Tics: (Conceptos, características, principios).

5° Momento.-Cada grupo escogerá un representante para socializar en plenaria de curso los conocimientos adquiridos en el desarrollo de la investigación realizada, y seguido se realizara un debate enfocado en el tema.

3° Proceso Pedagógico: Apropiación del conocimiento (Conceptualización y Teorización):

6° Momento.- El instructor formador planteara actividades investigativas sobre los siguientes temas:

- algoritmos de programación
- diseño de diagrama de flujo
- formulación de pseudocodigos
- estructuras condicionales
- estructuras cíclicas
- manejo de conceptos de la POO como: clase, objetos, métodos, herencia, polimorfismo, encapsulamiento.

Las cuáles serán expuestas en clases, profundizadas y corregidas por el instructor.

7° Momento.- El instructor formador realizará actividades prácticas sobre

- Creación de algoritmos en diagramas de flujo y pseudocodigo
- Implementación de estructuras condicionales en algoritmos
- Implementación de estructuras cíclicas en algoritmos
- Implementación de conceptos de la POO en algoritmos

Las cuáles serán revisadas, corregidas y explicadas por el instructor en formación.

4° Proceso Pedagógico: Transferencia del conocimiento:

8° Momento.-Con los propósitos de adquirir las habilidades y destrezas, y obtener la evidencia requerida para la evaluación, los alumnos en compañía del instructor formador demostraran el manejo de conceptos asociados a la POO como por ejemplo algoritmos, condicionales, ciclos, etc a través de socializaciones grupales y presentación de pruebas de conocimiento.

- Ambiente (s) Requerido (s):

1°.- Sala convencional de Informática dotada con 30 equipos de cómputo de escritorio y/o portátiles con conexión a Internet, 20 mesas trapezoidales y 40 sillas unipersonales sin brazo, Tablero de acrílico, Video beam, Aire acondicionado.

- Materiales de consumo :

- Marcador borrable (2).
- Lapicero (1).
- Lápiz (1).
- Borrador para tablero acrílico (1).

4. ACTIVIDADES DE EVALUACIÓN

Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
Evidencias de Conocimiento : Maneja los algoritmos, lógica y técnicas asociadas a la programación orientada a objetos necesarias para interpretar un diseño. **Evidencias de Desempeño:** **Evidencias de Producto:**	1-ELABORA PROPUESTAS DE TRABAJO, DE ACUERDO CON LA INTERPRETACIÓN DE LAS NECESIDADES TECNOLÓGICAS, EXPUESTAS EN EL INFORME DE REQUERIMIENTOS, SEGÚN NORMAS Y PROTOCOLOS DE LA EMPRESA. 2-REPRESENTA PROCESOS DEL SISTEMA A PARTIR DE LA CONSTRUCCIÓN DE ALGORITMOS, COMO PARTE DE LA SOLUCIÓN A SITUACIONES PLANTEADAS, UTILIZANDO LENGUAJES DE PROGRAMACIÓN ORIENTADOS A OBJETOS.	**Técnica:** **Valoración del conocimiento** **Instrumento de Evaluación:** **Prueba de conocimiento** **Técnica:** **Instrumento de Evaluación:** **Técnica:** **Instrumento de Evaluación:**

5. GLOSARIO DE TERMINOS:

1. Abstracción: Propiedad y/o técnica de software que oculta los detalles de la implementación. Java soporta abstracción de clases y abstracción de métodos. La abstracción de métodos se define separando el uso de un método sin conocer como está implementado ese método. Si decide combinar la implementación, el programa cliente será afectado. De modo similar la abstracción de clases oculta la implementación de la clase del cliente.
2. Algoritmo: Método que describe cómo se resuelve un problema en término de las acciones que se ejecutan y especifica el orden en que se ejecutan estas acciones. Los algoritmos ayudan al programador a planificar un programa antes de su escritura en un lenguaje de programación.
3. Análisis: Proceso de identificación, modelado y descripción de lo que hace un sistema y de cómo trabaja.
4. Aplicación: Programa autónomo Java tal como cualquier programa escrito utilizando un lenguaje de alto nivel. Las aplicaciones se pueden ejecutar desde cualquier computadora con un intérprete Java. Las aplicaciones no están sometidas a las restricciones impuestas los appletsdeJava. Una clase aplicación debe contener un método main. Se utiliza como sinónimo de programa.
5. Argumento: Información pasada a un método. Los argumentos se suelen llamar también parámetros. Un método que espera recibir argumentos debe contener una declaración de argumentos formales por cada argumento actual como parte de la cabecera del mismo. Cuando se invoca a un método, los valores de los argumentos actuales 8reales) se copia en los correspondientes argumentos formales. Véase parámetro actual (actual parameter).
6. Asignación: Almacenamiento de un valor en una variable. La sentencia de asignación es aquella que implementa la asignación y utiliza un operador de asignación.
7. Biblioteca de clases: Colección organizada de clases que proporciona un conjunto de componentes y abstracciones reutilizables.
8. Binario: Representación numérica en base 2. En esta base sólo se utilizan los dígitos 0 y 1. Las posiciones de los dígitos representan potencias sucesivas de 2. Véase bit.
9. Bolean: Tipos primitivos de datos en Java. El tipo bolean puede tomar sólo dos valores: true (verdadero) y false (falso).
10. Clase: Colección encapsulada de datos y operaciones que actúan sobre los datos. El concepto de clase es fundamental en programación orientada a objetos. Una clase consta de métodos y datos. Los métodos de una clase definen el conjunto de operaciones permitidas sobre los datos de una clase (sus atributos). Una clase puede tener muchas instancias de la clase u objetos.
11. Encapsulamiento, encapsulación: Localización y protección de las características internas y estructura de un objeto. Combinación de métodos y datos en una única estructura de datos. En Java se conoce como clase.
12. Herencia: (inheritance) Una relación entre clases en que una subclase se extiende desde una superclase.

6. REFERENTES BILBIOGRAFICOS

- Bases de programación**.** **https://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=28&Itemid=59**
- Introducción a la algoritmia - **https://codigofacilito.com/cursos/introduccion**

7. CONTROL DEL DOCUMENTO

	Nombre	**Cargo**	**Dependencia**	**Fecha**
Autor (es)	Sneider David Pineda Sánchez, Arturo Miranda Lengua, Andrés Ramos Morales, José miguel Dager Montoya	Instructor	Centro de Comercio, Industria y Turismo	Marzo 08 de 2019

8. CONTROL DE CAMBIOS (diligenciar únicamente si realiza ajustes a la guía)

	Nombre	**Cargo**	**Dependencia**	**Fecha**	**Razón del Cambio**
Autor (es)					