1. IDENTIFICACIÓN DE LA GUIA DE APRENIZAJE **Guía N° 03**

- **Denominación del Programa de Formación**: Programación de Software
- **Código del Programa de Formación**: 228120
- **Nombre del Proyecto:** Desarrollo de software para el mejoramiento de los procesos académicos - administrativos en las instituciones educativas articuladas.
- **Fase del Proyecto:** Planeación
- **Actividad de Proyecto:** planear el desarrollo del software mediante el diseño de base de datos e interfaz de usuario.
- **Competencia**: Analizar los requisitos del cliente para construir el sistema de información.
- **Resultados de Aprendizaje Alcanzar:** Interpretar los diagramas de caso de uso, de objetos, de estado, de secuencia, de paquetes o componentes, de despliegue, de colaboración según el diseño entregado
- **Resultados de Aprendizaje Alcanzar:** Describir que son los modificadores para aplicarlos a un proyecto de formación
- **Duración de la Guía:** 130 Horas

2. PRESENTACION



Es muy importante destacar que cuando hacemos referencia a la programación orientada a objetos no estamos hablando de unas cuantas características nuevas añadidas a un lenguaje de programación. Estamos hablando de una nueva forma de pensar acerca del proceso de descomposición de problemas y de desarrollo de soluciones de programación. La programación orientada a objetos surge en la historia como un intento para dominar la complejidad que, de forma innata, posee el software. Tradicionalmente, la forma de enfrentarse a esta complejidad ha sido empleando lo que llamamos programación estructurada, que consiste en descomponer el problema objeto de resolución en sub-problemas y más sub-problemas hasta llegar a acciones muy simples y fáciles de codificar. Se trata de descomponer el problema en acciones, en verbos. En el ejemplo de un programa que resuelva ecuaciones de segundo grado, descomponíamos el problema en las siguientes acciones: primero, pedir el valor de los coeficientes a, b y c; después, calcular el valor de la discriminante; y por último, en función del signo del discriminante, calcular ninguna, una o dos raíces.

Como podemos ver, descomponíamos el problema en acciones, en verbos; por ejemplo, el verbo pedir, el verbo hallar, el verbo comprobar, el verbo calcular… La programación orientada a objetos es otra forma de descomponer problemas. Este nuevo método de descomposición es la descomposición en objetos; vamos a fijarnos no en lo que hay que hacer en el problema, sino en cuál es el escenario real del mismo, y vamos a intentar simular ese escenario en nuestro programa.

Nota: Se debe crear un glosario donde de condensen todas las palabras desconocidas que encuentren durante el desarrollo de la guía.

¡Bienvenidos al mundo de la programación!

3. FORMULACION DE LAS ACTIVIDADES DE APRENDIZAJE

3.1 **Actividades de Reflexión inicial.**



Observar la siguiente presentación:

"Orientación a Objetos" (el documento se encuentra disponible en la *carpeta material de apoyo*)

Una vez vista la presentación, formar grupos de 3 aprendices y realizar un conversatorio donde se difieran los diferentes puntos de vista mediante el pensamiento crítico, se deben sacar por lo menos 5 conclusiones por grupo.

- **Duración**: 2 horas
- **Ambiente Aprendizaje**: Sala de Sistemas
- **Materiales:** Software aplicado

3.2 Actividades de contextualización e identificación de conocimientos necesarios para el aprendizaje.

De respuesta a los siguientes interrogantes:

Resalte con color amarillo sobre la letra según su criterio: "V" si es verdadero y "F" si es falso:

- POO es programación orientada a objetos (V) (F)
- Una Clase es la agrupación de objetos con determinadas características (V) (F)
- Una método son las acciones que identifican a una clase (V) (F)
- Un atributo son las características asociadas a un objeto (V) (F)
- Una clase puede heredar los atributos de otra clase (V) (F)
- Una clase abstracta es aquella donde se encuentran diferentes objetos que tiene capacidad de hacer cosas diferentes pero estos pertenecen a la misma clase (V) (F)
- Encapsulamiento es una propiedad que permite poner seguridad a una clase (V) (F)
- Una clase abstracta es aquella que es fiel copia de otra incluyendo todos sus métodos y atributos (V) (F)
- Un objeto es un programa (V) (F)
- Una clase polimórfica es aquella que soporta objetos con diferentes métodos (V) (F)
- Una clase puede heredar de otra clase sus métodos (V) (F)

Cierre de Actividad:



Las respuestas deben consolidar en un documento Word y presentarlo al docente/instructor.

- **Duración**: 1 horas
- **Ambiente Aprendizaje**: Sala de Sistemas
- **Materiales:** Software aplicado

3.3 Actividades de apropiación del conocimiento (Conceptualización y Teorización).

3.3.1 Interpretar Elaborar los diagramas UML que representen las necesidades del cliente.

Identificar los componentes principales y la estructura de la programación orientada a objetos, para ello debe dirigirse a la carpeta material de apoyo y ver el video llamado "Introduccin_a_POO_Programacion_Orientada_a_Objetos" o también lo pueden ver online mediante el siguiente link: https://www.youtube.com/watch?v=xVqj2xe9cK0. Se deben formar grupos de 3 aprendices y tomar 5 ejemplos de la vida cotidiana y aplicar todo el paradigma de la programación orientada a objetos.

Cada grupo debe presentar 5 ejemplos propuestos, plasmándolos en una herramienta case y exponerlos, el docente técnico/instructor, aclarara dudas y retroalimentará a los grupos.

Para el desarrollo de esta actividad, pueden consultar el documento de apoyo llamado "Ejemplo_POO.pdf" ubicado en la carpeta material de apoyo.

- **Duración**: 20 horas
- **Ambiente Aprendizaje**: Sala de Sistemas
- **Materiales:** Software aplicado

3.3.2 Reconocer los conceptos principales de la programación orientada a objetos.

De acuerdo con lo estudiado y realizado en los puntos anteriores y apoyándose en el documento llamado "Explicación_POO.pdf" el cual se encuentra en el material de apoyo. Con sus propias palabras responda las siguientes preguntas:

- Objeto:
- Clase:
- Atributo:
- Abstracción:
- Encapsulamiento:
- Modularidad:
- Generalización:
- Persistencia:
- Operación:
- Herencia:
- Polimorfismo:

Las preguntas deben ser respondidas mediante el cuestionario llamado "Cuestionario_Conceptos_POO" (el documento referido se encuentra disponible en la carpeta instrumentos de evaluación).

- **Duración**: 2 horas
- **Ambiente Aprendizaje**: Sala de Sistemas
- **Materiales:** Software aplicado

3.3.3 Conocer la relación que existe la programación orientada a objetos (POO)

El lenguaje modelado unificado(UML) y la importancia que tienen dentro del desarrollo de software, para ello nos centraremos en analizar el diagrama de objetos y su la función que cumple. Leer y apropiar el siguiente material de apoyo:

4 POO – UML.pdf (página 41)
5 Diagramas_del_UML.pdf (Diagrama de Objetos)

En grupo de 3 aprendices hacer un análisis crítico sobre el diagrama de objetos, finalmente se deben preparar y presentar 6 ejemplos de la vida cotidiana donde se evidencie claramente la relación entre los objetos. En esta actividad se puede utilizar cualquier herramienta case para el modelado.

- **Duración**: 5 horas
- **Ambiente Aprendizaje**: Sala de Sistemas
- **Materiales:** Software aplicado

3.3.4 A continuación se relacionan 6 casos de estudio, uno de los cuales le será asignado por el Docente/instructor a su equipo de trabajo.

Usted procederá a:
Construir una presentación donde muestre y explique claramente el diagrama de objetos aplicado al caso asignado, en esta actividad deben continuar los grupos previamente conformados.

- Sistema Para El Manejo De Ventas e Inventarios De Una Empresa
- Sistema Para Alquiler, Reserva y Venta De Películas
- Sistema Para Una Agencia De Viajes
- Sistema Para El Registro De Notas De Un Colegio
- Sistema Para Un Parqueadero Publico
- Sistema Para Empresa De Prestamos

Los grupos realizaran diferentes consultas y en varias fuentes de información o hacer visitas de campo, con el fin de hacer un buen levantamiento de información que sirva como insumo para la creación del diagrama.
Se deben entregar los siguientes diagramas:

- Casos De Uso
- Clases
- Objetos

Para esta actividad se debe utilizar cualquier herramienta case.

Cierre De Actividad:
Cada grupo deberá preparar la exposición y presentar el trabajo realizado, el docente técnico/instructor moderará dicha exposición.

- **Duración**: 50 horas
- **Ambiente Aprendizaje**: Sala de Sistemas
- **Materiales:** Software aplicado

Tomando como referencia la planeación pedagógica y las orientaciones para elaborar guías de aprendizaje citado en la guía de desarrollo curricular

4. ACTIVIDADES DE EVALUACIÓN

3.4 Actividades de transferencia del conocimiento.
Planear el desarrollo del software mediante el diseño de base de datos e interfaz de usuario.

3.4.1 En esta actividad se deben crear los siguientes diagramas del proyecto, teniendo en cuenta los requerimientos levantados en la fase de análisis y entregarlos en la herramienta case seleccionada:

- Casos De Uso
- Diagrama De Objetos
- Diagrama De Clases

Descripción De La Actividad:
Se debe continuar con la misma dinámica de los grupos previamente conformados, la evidencia debe entregarse al docente/instructor en medio digital.

- **Duración**: 50 horas
- **Ambiente Aprendizaje**: Sala de Sistemas
- **Materiales:** Software aplicado

Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
Evidencias de Conocimiento: 3.3.2 Documento en Word responder encuesta en plataforma. Reconocer los conceptos principales de la programación orientada a objetos.	-Modela la base de datos, a partir de la valoración de la información obtenida en el diccionario de datos y el análisis de los procesos, de acuerdo con las necesidades del sistema de información requerido.	**Cuestionario**
Evidencias de Desempeño: 3.3.1 Exposición ejemplos case. Identificar los componentes	- Realiza el informe de análisis del sistema de información	**Lista de chequeo Desempeño**

principales y la estructura de la programación orientada a objetos 3.3.3 archivo para presentar el modelado. Conocer la relación que existe la POO, el UML y la importancia dentro del desarrollo de software 3.3.4 Presentar una exposición que aplique la actividad Diagramas Casos De Uso, Clases y Objetos.	requerido, presentando el estado actual de la organización, y proponiendo alternativas de solución, de acuerdo con las normas y protocolos establecidos. Utiliza herramientas case para elaborar diagramas de casos de uso, que representen el estado actual de los componentes del sistema, apoyado en el análisis del informe de requerimientos. Elabora los diagramas UML, de acuerdo con las características de cada uno de ellos, basado en los requerimientos del cliente, utilizando herramientas case.	
Evidencias de Producto: 3.4.1 Elaborar los diagramas UML, Crear el diagrama de casos de uso, clases y diagramas de objetos de la idea de negocio.		**Lista de chequeo Producto**

5. GLOSARIO DE TERMINOS

Clase: Conjunto de objetos que comparten una estructura y un comportamiento común
Herencia: Técnica para permitir a las clases usar los métodos y los datos de una clase padre
Modelo: Abstracción de algo con el propósito de entenderlo, posiblemente antes de construirlo.
Objeto: Representa una entidad ya sea física, conceptual o software
POO: Programación Orientada a Objetos
UML: El lenguaje de modelado unificado
Lenguaje de Programación: Es un lenguaje formal diseñado para expresar procesos que pueden ser llevados a cabo por máquinas como las computadoras.
MER: Modelo Entidad Relación.
DER: Diseño Entidad Relación

6. REFERENTES BILBIOGRAFICOS

- Khoshafian, S. Abnous, R. Object Orientation. John Wiley & Sons. 1995.
- Korson, T. McGregor, J. Understanding Object Oriented: a Unifying Paradigm. Communications of the ACM. 33(9). 9/1990.
- Eckel, B. Thinking in Java. Prentice-Hall. 1998.
- Gamma, E. Helm, R. Design Patterns. Addison-Wesley, 1995.
- The Java Tutorial. http://java.sun.com/
- Lalonde, W. Pugh, J. Inside Smalltalk, Vol. I y II. 05/1994.
- Oestereich, Bernd. "Developing Software with UML". Addison-Wesley, 1997.

7. CONTROL DEL DOCUMENTO

	Nombre	**Cargo**	**Dependencia**	**Fecha**
Autor (es)	GUSTAVO ADOLFO GRUESO	INSTRUCTOR	CEAI	Agosto 2015

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

	Nombre	**Cargo**	**Dependencia**	**Fecha**	**Razón del Cambio**
Autor (es)	GUSTAVO ADPOLFO GRUESO	INSTRUCTOR	CEAI	Abril 2017	Cambio de formato