1. IDENTIFICACIÓN DE LA GUIA DE APRENIZAJE

- Denominación del Programa de Formación: **PROGRAMACIÓN 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): **ANÁLISIS**
- 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**
- Resultados de Aprendizaje Alcanzar: **437635 - 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**
- Duración de la Guía: **60 horas.**

2. PRESENTACION

- Motivación hacia la actividad de aprendizaje:

Lectura de la Introducción:

UML (Unified Modeling Language) es un lenguaje que permite modelar, construir y documentar los elementos que forman un sistema software orientado a objetos. Se ha convertido en el estándar de facto de la industria, debido a que ha sido concebido por los autores de los tres métodos más usados de orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh. Estos autores fueron contratados por la empresa Rational Software Co. para crear una notación unificada en la que basar la construcción de sus herramientas CASE. En el proceso de creación de UML han participado, no obstante, otras empresas de gran peso en la industria como Microsoft, Hewlett-Packard, Oracle o IBM, así como grupos de analistas y desarrolladores.

UML usa elementos y los asocia de diferentes formas para formar diagramas que representan aspectos estáticos o estructurales de un sistema, y diagramas de comportamiento, que captan los aspectos dinámicos de un sistema.

En esta práctica es común crear especificaciones suplementarias para capturar detalles de requisitos que caen fuera del ámbito de las descripciones de los casos de uso. Ejemplos de esos temas incluyen restricciones de diseño como: rendimiento, temas de escalabilidad/gestión, o cumplimiento de estándares.

HERRAMIENTAS CASE: Computer Aided Software Engineering, Ingeniería de Software Asistida por Computadora. Son diversas aplicaciones informáticas o programas informáticos destinadas a aumentar la

productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero.

OBJETIVOS HERRAMIENTAS CASE:

- Mejorar la productividad del software.
- Aumentar la calidad del software.
- Reducir el tiempo y costo de desarrollo y mantenimiento de los sistemas informáticos.
- Mejorar la planificación de un proyecto.
- Aumentar la biblioteca de conocimiento informático de una empresa ayudando a la búsqueda de soluciones para los requisitos.
- Automatizar el desarrollo del software, la documentación, la generación de código, las pruebas de errores y la gestión del proyecto.
- Ayuda a la reutilización del software, portabilidad y estandarización de la documentación.
- Gestión global en todas las fases de desarrollo de software con una misma herramienta.
- Facilitar el uso de las distintas metodologías propias de la ingeniería del software.

CLASIFICACIÓN HERRAMIENTAS CASE:

- Upper CASE (U-CASE): Herramientas que ayudan en las fases de planificación, análisis de requisitos y estrategia del desarrollo, usando, entre otros diagramas UML.
- Middle CASE (M-CASE): Herramientas para automatizar tareas en el análisis y diseño de la aplicación.
- Lower CASE (L-CASE): Herramientas que semi-automatizan la generación de código, crean programas de detección de errores, soportan la depuración de programas y pruebas. Además automatizan la documentación completa de la aplicación. Aquí pueden incluirse las herramientas de desarrollo rápido de aplicaciones.

Mediante el desarrollo de esta guía de aprendizaje, los alumnos podrán adquirir las habilidades y destrezas para ser competentes a la hora de interpretar diagramas de casos de uso, diagrama de clases , diagramas de transición de estados, diagrama de secuencias, necesarios para representar el modelo del sistema; así cumpliendo de esta manera con los criterios de evaluación establecidos en los diseños curriculares del Servicio Nacional de Aprendizaje para la titulación de Técnico en programación de Software.

Video relacionado: Los aprendices en compañía del Instructor visualizarán un video el cual tiene como título “HERRAMIENTAS CASE”, que tiene entre su temática relacionada los casos de usos en UML, Este estará en la ubicación MIS DOCUMENTOS del equipo de cómputo asignado.

- Guía y organización del aprendizaje: El instructor 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:

Cada grupo de aprendices escogerá un representante para realizar la socialización de trabajos realizados, permitiendo la participación activa de todos sus compañeros del curso.

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

Con los propósitos de adquirir las habilidades y destrezas, y obtener la evidencia requerida para la evaluación, los alumnos conformados en grupos de 4 personas realizarán en compañía del Docente Formador ACTIVIDADES DE TIPO PRÁCTICO SEGÚN REQUERIMIENTOS CURRICULARES.

3. FORMULACION DE LAS ACTIVIDADES DE APRENDIZAJE

- Descripción de la(s) Actividad(es):

Actividad de Enseñanza: Aprendizaje; Evaluación #1:

DISEÑAR DIAGRAMAS UML DE ACUERDO AL DISEÑO ENTREGADO USANDO HERRAMIENTAS CASE. Tiempo máximo: 60 horas.

1° Proceso Pedagógico: Actividad de Reflexión inicial:

1° Momento.- Los aprendices en compañía del instructor realizarán la lectura de la Introducción y observarán criticamente el vídeo propuesto.

2° Proceso Pedagógico: Contextualización e identificación de conocimientos:

2° Momento.- Los aprendices en compañía con el instructor realizarán la socialización en plenaria de los conocimientos adquiridos en el momento #1.

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

3° Momento.- El instructor resolverá las dudas de los aprendices surgidas de la realización de los momentos anteriores.

4° Momento.- El docente formador realizará charlas magistrales donde expondrá con claridad a los alumnos los diversos temas relacionados con:

- UML
- DISEÑO DE DIAGRAMAS DE CASOS DE USO
- DISEÑO DE DIAGRAMAS DE OBJETOS
- DISEÑO DE DIAGRAMAS DE MÁQUINA DE ESTADOS
- DISEÑO DE DIAGRAMAS SECUENCIA
- DISEÑO DE DIAGRAMAS DE CLASES
- DISEÑO DE DIAGRAMAS DE PAQUETES
- DISEÑO DE DIAGRAMAS DE DESPLIEGUE
- DISEÑO DE DIAGRAMAS DE COLABORACIÓN
- MANEJO DE HERRAMIENTAS CASE (ARGOUML, MINDOMO, CANVA, GOCONQR, ENTRE OTRAS)

4° Proceso Pedagógico: Transferencia del conocimiento:

5° Momento.- Los aprendices conformados en grupos de 3 personas realizarán los diagramas de la aplicación a desarrollar, utilizando aplicaciones en línea actuales tales como Mindomo, Canva, GoConqr, entre otros. Se debe compartir el enlace con instructor para su valoración final.

- Ambiente (s) Requerido (s):

1°. - Sala convencional de Informática dotada con 30 equipos de cómputo de escritorio y/o portátiles con conexión a Internet, 30 escritorios y 40 sillas unipersonales sin brazo, Tablero de acrílico, Video beam, Fotocopiadora, Aire acondicionado.
2°. - Aula convencional para sesiones de FPI dotada con 40 sillas unipersonales con brazo, Tablero de acrílico, Video beam, ventiladores de techo, escritorio y silla para el docente.

- Materiales de consumo :

- Papel bond tamaño carta (1).
- Marcador borrable (2).
- Lapicero (1).
- Lápiz (1).
- Borrador para tablero acrílico (1).
- Tablilla de anotaciones (1).
- Memoria USB (1).

4. ACTIVIDADES DE EVALUACIÓN

Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
Evidencias de Conocimiento : **Evidencias de Desempeño:** **Evidencias de Producto:** Diseño de diagrama de casos de uso y diagramas UML necesarios, para representar el modelado del sistema, basándose en el análisis de los requerimientos del cliente y utilizando herramientas CASE.	ELABORA LOS DIAGRAMAS UML, DE ACUERDO CON LAS CARACTERÍSTICAS DE CADA UNO DEELLOS, BASADO EN LOS REQUERIMIENTOS DEL CLIENTE, UTILIZANDO HERRAMIENTAS CASE.	**Técnica:** **Instrumento de Evaluación:** **Técnica:** **Instrumento de Evaluación:** **Técnica:** Valoración de Informe. **Instrumento de Evaluación:** Lista de chequeo o verificación.

5. GLOSARIO DE TERMINOS:

HERRAMIENTAS CASE: *Computer Aided Software Engineering*, Ingeniería de Software Asistida por Computadora. Son diversas aplicaciones informáticas o programas informáticos destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero.
DIAGRAMAS DE CASOS DE USO: se usan para modelar las interacciones del usuario/sistema. Ellos definen el comportamiento, requisitos y restricciones en la forma de scripts o escenarios.
DIAGRAMAS DE OBJETOS: muestran como las instancias de elementos estructurales se relacionan y usan en tiempo de ejecución.
DIAGRAMAS DE MÁQUINA DE ESTADOS: son esenciales para el entendimiento de la condición momento a momento o "estado de ejecución" de un modelo cuando éste se ejecuta
DIAGRAMAS SECUENCIA: están estrechamente relacionados a los diagramas de comunicaciones y muestran la secuencia de mensajes pasadas entre los objetos usando una línea de tiempo vertical.
DIAGRAMAS DE CLASES: definen los bloques de construcción básica de un modelo: los materiales generales, clases y tipos que se usan para construir un modelo completo.
DIAGRAMAS DE PAQUETES: se usan para dividir el modelo en contenedores lógicos o "paquetes" y describen las interacciones entre ellos a un nivel más alto.
DIAGRAMAS DE DESPLIEGUE: muestran la disposición de artefactos significativos dentro de una configuración real.
DIAGRAMAS DE COLABORACIÓN: muestran la red y la secuencia de mensajes de comunicaciones entre objetos en tiempo de ejecución durante una instancia de colaboración.
ARGOUML: es una aplicación de diagramado de UML escrita en Java y publicada bajo la Licencia EPL.

6. REFERENTES BILBIOGRAFICOS

- Modelado en Sistemas UML. Popkin Software and Systems. Centro de Recursos de UML en www.popkin.com
- Entendiendo UML: La guía del desarrollador, con una aplicación java basada en web, por Paul Harmon y Mark Watson; Morgan Kauffman Publishers, Inc., 1998 (www.mkp.com/books_catalog/1-55860-465-0.asp).
- ¿Qué es el lenguaje unificado de modelado (UML)?: https://www.lucidchart.com/pages/es/que-es-el-lenguaje-unificado-de-modelado-uml
- Manual de UML. Paul Kimmel. MacGraw Hill Profesional.
- Tutorial UML 2.3: http://www.sparxsystems.com.ar/resources/uml2_tutorial.html

7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	José Miguel Dager, Sneider Pineda Sánchez , Andrés Ramos Morales, Arturo Miranda Lengua	Instructor	Centro de Comercio, Industria y Turismo	Abril 2 de 2019

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

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					