**1. IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE**

* **Denominación del programa de formación:** ANÁLISIS Y DESARROLLO DE SISTEMAS DE INFORMACIÓN
* **Código del programa de formación:** 228106
* **Nombre del proyecto:** DESARROLLO E IMPLEMENTACIÓN DE SOFTWARE DE ACUERDO LAS NECESIDADES DEL SECTOR TEXTIL EN ANTIOQUIA
* **Fase del proyecto:** PLANEAR
* **Actividad de proyecto:** DEFINIR Y ANALIZAR LOS REQUERIMIENTOS DEL SOFTWARE A DESARROLLAR APLICANDO EL ESTÁNDAR DE CALIDAD
* **Competencia:** 
  + 220501032 - ANALIZAR LOS REQUISITOS DEL CLIENTE PARA CONSTRUIR EL SISTEMA DE INFORMACIÓN
* **Resultados de aprendizaje alcanzar**:
  + REPRESENTA EL BOSQUEJO DE LA SOLUCIÓN AL PROBLEMA PRESENTADO POR EL CLIENTE, MEDIANTE LA ELABORACIÓN DE DIAGRAMAS DE CASOS DE USO, APOYADO EN EL ANÁLISIS DEL INFORME DE REQUERIMIENTOS, AL CONFRONTAR LA SITUACIÓN PROBLÉMICA CON EL USUARIO SEGÚN NORMAS Y PROTOCOLOS DE LA ORGANIZACIÓN
* **Duración de la guía:** 60 HORAS (2 SEMANAS) PRESENCIALES - 16 HORAS (0.5 SEMANAS) AUTÓNOMAS
  + - FECHA INICIO:
    - FECHA FIN:

**2. PRESENTACIÓN**

Estimado aprendiz:

La presente guía de aprendizaje le entregará un conjunto de saberes y haceres relacionados al análisis de requerimientos de software, aplicando los conceptos básicos del lenguaje de modelado universal (UML), además de reconocer el rol del analista en el desarrollo de sistemas de información. Esto, partiendo de sus saberes previos, sin duda alguna le permitirá desarrollar habilidades integrales asociadas a su formación profesional permitiéndole ser autónomo, sistemático y organizado en el desarrollo de cada una de las tareas asignadas, de igual modo se espera que tenga un papel activo, propositivo y colaborativo con el resto de participantes del proceso de formación.

Recuerde que tendrá a su disposición los recursos y materiales didácticos correspondientes según los requerimientos de la presente actividad de aprendizaje, entre ellos un equipo de instructores capacitados y dispuestos a atender cualquier inquietud que surja como consecuencia del desarrollo de la presente guía.

**3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE**

* **Descripción de la(s) Actividad(es):** Las siguientes actividades están diseñadas para brindar herramientas al aprendiz que permitan modelar sistemas de información a partir del análisis de los requerimientos obtenidos en etapas previas. El aprendiz dimensionará la importancia del analista en el proceso de desarrollo de sistemas de información.
* **Ambiente Requerido:** Aula convencional, ambiente de aprendizaje (sala de sistemas), laboratorio práctico de sistemas, unidad productiva del entorno e infraestructura institucional SENA.
* **Materiales:** Mesas, sillas, computador con conexión a internet, cable HDMI, cable VGA, marcadores borrables, borrador de tablero, hojas de papel, bolígrafos.

**3.1 Actividades de reflexión inicial:**

* **Descripción de las actividades:**  Con el objetivo de conocer su apreciación sobre una situación expuesta y tener un primer acercamiento a la importancia del análisis de requerimientos en el desarrollo de sistemas de información, planteamos las siguientes acciones:

**Acción 1:** El rol del analista

Luego de un arduo trabajo interactuando con las bases de la lógica de programación, los algoritmos, y descubrir los engranajes que guían el funcionamiento de los sistemas de información, un aprendiz del programa de Análisis y Desarrollo de Sistemas de Información a quien llamaremos Jaime, conoce a un amigo que actualmente es desarrollador de software y tiene una empresa. Este amigo, al que llamaremos Santiago, le cuenta que ha desarrollado un par de plataformas con un grupo de trabajo dotado de alta pericia en lenguajes de programación y bases de datos. Jaime, indica que ha aprendido a desarrollar algoritmos y se siente en el escalón más alto de la cadena evolutiva, a lo que Santiago le indica que los algoritmos son importantes, pero que en muchas ocasiones no son lo más necesario en el desarrollo del proyecto. Santiago manifiesta que ha tenido dificultades en la creación de productos de software con sus clientes y su equipo de trabajo, y le plantea las siguientes preguntas a su amigo aprendiz:

1. ¿Qué actividades se deben hacer antes codificar la solución en un lenguaje de programación?
2. ¿A quién van dirigidos los sistemas de información?
3. ¿Por qué muchos proyectos se entregan después del tiempo estimado?, incluso, ¿Por qué algunos no se terminan?
4. ¿Cuáles considera que son los principales problemas detectados en los equipos de trabajo en el área de desarrollo de software?
5. ¿Es realmente importante el ciclo de vida del software?
6. ¿Por qué no debería empezar a programar desde el momento en que el cliente indica una necesidad?
7. ¿Por qué no hacerlo, si ese es “el sabor”?
8. ¿Por qué los clientes no entienden lo que les quiero decir cuando presentamos una solución?
9. ¿Qué pasa cuando inicias un proyecto y no lo planeas
10. ¿Qué podría pasar si construimos un edificio sin planos?

El instructor asumirá el rol de Santiago, y realizará las preguntas anteriormente citadas a los aprendices, quienes asumirán el rol de Jaime, para así reconocer sus percepciones entorno al caso de estudio.

**3.2 Actividades de contextualización e identificación de conocimientos necesarios para el aprendizaje:**

* **Descripción de las actividades:** En esta etapa se acercará al aprendiz al contexto de las funciones que cumplen los analistas de sistemas en el desarrollo de los sistemas de información. Para cumplir este objetivo, el aprendiz debe desarrollar las siguientes acciones:

**Acción 1:** En busca del analista

En aras de impulsar la motivación en los aprendices, vamos a realizar la siguiente actividad en la cual se identificará la importancia de ser una analista. Para esta actividad, cada aprendiz debe realizar búsquedas de ofertas en portales de trabajo con los siguientes parámetros de búsqueda:

1. Analista de sistemas

2. Analista desarrollador

3. Analista funcional de sistemas

4. Analista desarrollador

5. Analista funcional + UML

Los resultados esperados pueden ser como estos:





Para esta actividad, cada aprendiz dispone de 20 minutos, luego de los cuáles debe dar respuesta a las siguientes preguntas:

1. ¿Sabe usted cuáles son las funciones y roles que cumple el analista de sistemas en un proyecto de tecnología?.
2. Escriba 5 actividades que realiza el analista en un proyecto de tecnología.
3. ¿Te animarías a ser analista funcional? Indica por qué

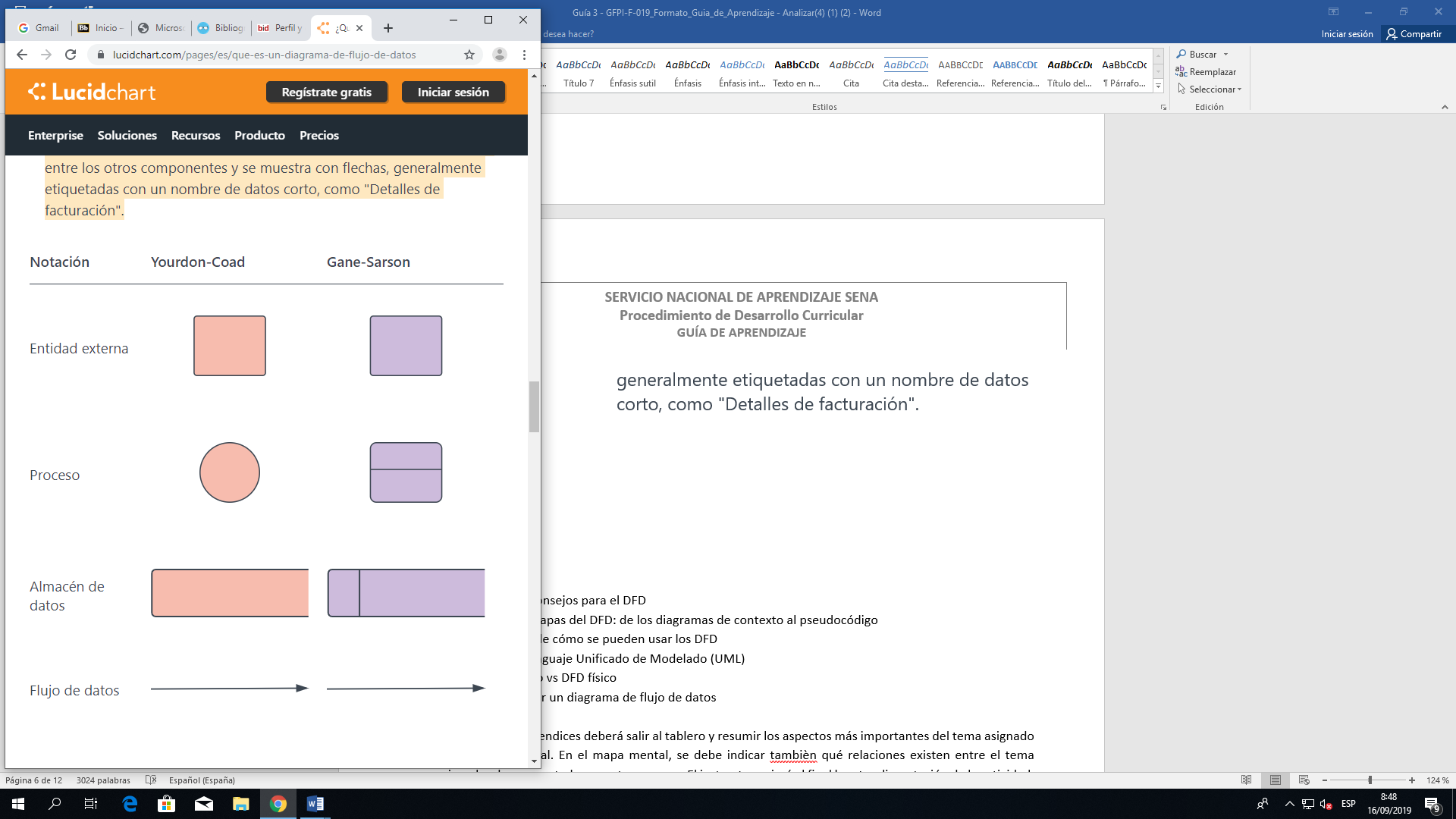
**Acción 2:** El universo del discurso

Uno de los factores determinantes para indicar el éxito de los proyectos de software se puede asociar al análisis a conciencia de los requerimientos identificados previamente y el modelado de la solución a construir. Una de las herramientas más empleadas para el análisis sons los DFD (diagramas de flujo de datos). Para esta acción, los aprendices deberán ingresar al enlace (<https://www.lucidchart.com/pages/es/que-es-un-diagrama-de-flujo-de-datos>) y leer todo el documento que se describe en él. Luego de esto, en grupos de 3 aprendices, se asignará alguno de los siguientes temas basados en el documento especificado.

* ¿Qué es un diagrama de flujo?
* Historia de DFD
* **Símbolos y notificaciones usadas en los DFD:**
* ourdon-Coad
* Yourdon-DeMarco
* Gane-Sarson

Una diferencia importante en sus símbolos es que Yourdon-Coad y Yourdon-DeMarco usan círculos para procesos, mientras que Gane y Sarson usan rectángulos redondeados.

* + - 1. **Entidad externa:** un sistema externo que envía o recibe datos, comunicándose con el sistema que se está diagramando. Son las fuentes y destinos de la información que entra o sale del sistema. Podría ser una organización o persona externas, un sistema de computadoras o un sistema de negocios.
      2. **Proceso:**cualquier proceso que cambia los datos y produce un resultado. Podría realizar cálculos u ordenar datos basados en una lógica o dirigir el flujo de datos en función de reglas de negocios. Se usa una etiqueta pequeña para describir el proceso, por ejemplo "Enviar pago".
      3. **Almacén de datos:** archivos o repositorios que conservan información para uso posteriores: una tabla de base de datos, "Pedidos”.
      4. **Flujo de datos:** la ruta que los datos toman entre las entidades externas, los procesos y los almacenes de datos. Representa la interfaz entre los otros componentes y se muestra con flechas, generalmente etiquetadas con un nombre de datos corto, como "Detalles de facturación".



* **Reglas y consejos para el DFD**
* Cada proceso debe tener al menos una entrada y una salida.
* Cada almacén de datos debe tener al menos una entrada y una salida de flujo de datos.
* Los datos almacenados en un sistema deben pasar por un proceso.
* Todos los procesos en un DFD pasan a otro proceso o almacén de datos.
* Los datos almacenados en un sistema deben pasar por un proceso.
* Niveles y capas del DFD: de los diagramas de contexto al pseudocódigo
* Ejemplos de cómo se pueden usar los DFD

**DFD en ingeniería de software:** enfocado hacia el desarrollo técnico, en el cual se realiza más investigación previa para llegar a la codificación.

**DFD en análisis de negocios:** Los analistas de negocios emplean los DFD para analizar los sistemas existentes y encontrar ineficiencias, la diagramación hace que se más comprensible el proceso y sirve para detectar partes que pasaron inadvertidas o no se comprendieron por completo.

**DFD en la reingeniería de procesos de negocios:** Los DFD se pueden usar para modelar un flujo de datos mejor y más eficiente a través de un proceso de negocios.

**DFD en el desarrollo ágil:** Los DFD se pueden usar para visualizar y comprender los requisitos de negocios y técnicos y planificar los siguientes pasos.

**DFD en estructuras de sistemas:** Cualquier sistema o proceso se puede analizar en un detalle progresivo para mejorarlo en aspectos tanto técnicos como no técnicos.

* DFD vs Lenguaje Unificado de Modelado (UML)
* DFD Lógico vs DFD físico
* Cómo crear un diagrama de flujo de datos

Cada grupo de aprendices deberá salir al tablero y resumir los aspectos más importantes del tema asignado en un mapa mental. En el mapa mental, se debe indicar tambièn qué relaciones existen entre el tema asignado y los presentados por otros grupos. El instructor guiará al final la retroalimentación de la actividad.

**3.3 Actividades de apropiación**

* **Descripción de las actividades:** Con el objetivo de fortalecer las orientaciones entregadas por el instructor y a la vez apropiarse de los temas, subtemas relacionados con la elaboración de los diagramas que definen el contexto del software (flujos de datos, casos de uso y clases), el aprendiz debe desarrollar las siguientes acciones:

**Acción 1: Diagrama de flujo de datos**

En esta actividad, los aprendices deben reunirse en sus grupo de proyecto con los integrantes de otro proyecto, deben escuchar la idea del proyecto y a partir de las ideas expuestas por el otro grupo, deben construir el diagrama de flujo de datos de nivel cero, uno y dos.

Esta actividad corresponde a una evidencia de desempeño y producto, los diagramas (diseñados en una herramienta, puede ser la sugerida en esta guía o de libre elección) deberán subirse en el enlace de la plataforma Blackboard habilitado por el instructor.

**Acción 2: Acercamiento a SCRUM**

Las metodologías de desarrollo de software permiten a los equipos de trabajo establecer la forma en que desarrollan sus proyectos y plantear estrategias para intervenir en las diferentes fases que componen el ciclo de vida del software.

En esta actividad, los aprendices tendrán su primer acercamiento a una de las metodologías de desarrollo ágil más populares en la industria, la metodología SCRUM. Cada aprendiz debe consultar en la carpeta *Recursos de formación* los siguientes documentos:

-La guía de SCRUM

-Manifiesto ágil

Igualmente, debe profundizar los temas inmersos en los documentos, abordando los siguientes componentes de la misma: roles de los participantes, entregas, historias de usuario, eventos, reuniones y artefactos.

Tras leer el material propuesto, los aprendices y el instructor realizarán un conversatorio en el cual se compartirán los hallazgos obtenidos y se darán conclusiones sobre el tema.

Finalmente, y con el objetivo de aplicar los conocimientos adquiridos, los aprendices se reunirán en los grupos de proyecto formativo para realizar las siguientes acciones:

1. Realizar un Daily Meeting

2. Definir los requerimientos del sistema

3. Documentar las reglas de negocio

Esta actividad corresponde a una evidencia de producto, las tres entregas SCRUM (formato libre) deberán subirse en el enlace de la plataforma Blackboard habilitado por el instructor.

**Acción 3: Diagrama de casos de uso, una mirada al negocio**

En esta actividad, el instructor realizará la introducción a los casos de uso y su importancia en la etapa de análisis de requerimientos en los sistemas de información. Se revisarán aspectos como su definición, uso, componentes y ejemplos. Luego de esto, los aprendices se reunirán en los grupos de proyecto para realizar los diagramas de casos de uso aplicados a los requerimientos funcionales de su proyecto, éstos deben ser coherentes con la propuesta de planteada en el documento ERS IEEE 830.

Para el diseño de los casos de uso, se sugiere el uso de herramientas CASE como StarUML, Dia y ArgoUML.

Los casos de uso planteados, serán incluídos en la actividad de transferencia de la guía en curso.

**Acción 4: Diagrama de clases, una mirada al dominio**

En esta actividad, el instructor dará una introducción al diagrama de clases y su importancia en la etapa de análisis de requerimientos en los sistemas de información. Se revisarán aspectos como su definición, uso, componentes y ejemplos. Luego de esto, el instructor asignará a cada aprendiz uno de los siguientes casos:

1. Gestión de trenes de las compañías ferroviarias
2. Gestión de oficinas y aparcamientos de recinto IND
3. Gestión de relaciones humanas en departamentos
4. Gestión de una liga de baloncesto

El análisis de requerimientos, es un momento de reflexionar, proponer y modelar soluciones, es por esto que un mismo problema puede ser analizado desde diferentes puntos de vista. Teniendo en cuenta lo anterior, cada aprendiz debe plantear una propuesta de diagrama de clases para el caso asignado, que permita modelar su estructura y relaciones entre los objetos del dominio. Para el diseño del diagrama de clases, se sugiere el uso de herramientas CASE como StarUML, Dia y ArgoUML.

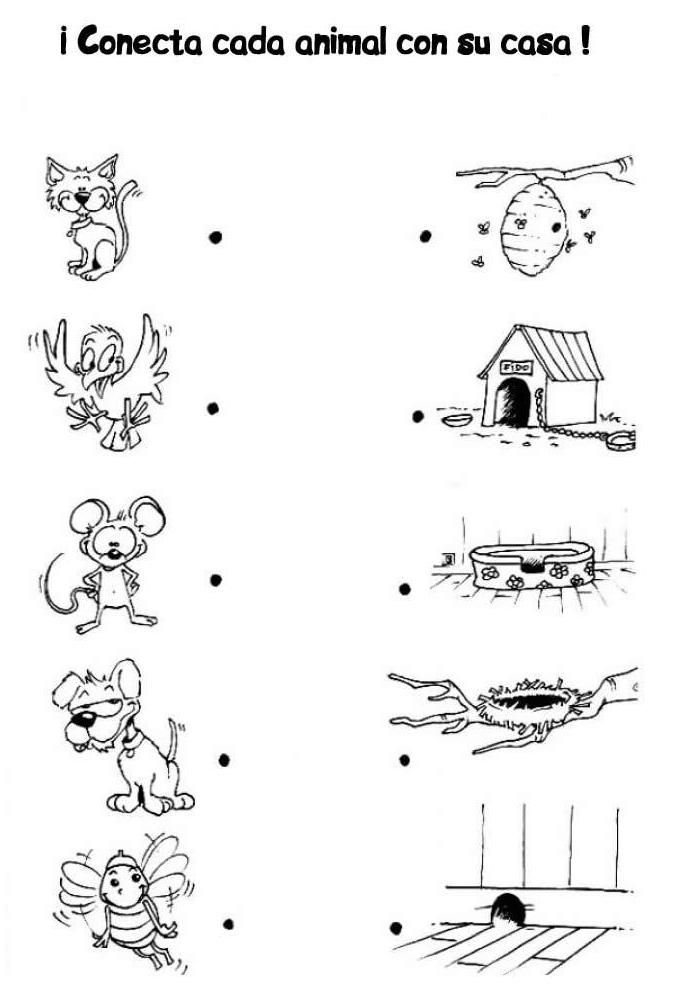
Al final, se socializarán algunas de las propuestas, para contrastarlas y establecer conclusiones del tema.

Esta actividad corresponde a una evidencia de conocimiento y producto,el diagrama de clases propuesto deberá subirse en el enlace de la plataforma Blackboard habilitado por el instructor.

**Acción 5: Diagramas UML**

En esta actividad, el instructor dará una perspectiva general de los tipos de diagramas UML (estructurales, comportamiento, interacción) en la etapa de análisis de requerimientos en los sistemas de información.

Seguidamente, y de manera individual, cada aprendiz investigará todos los diagramas UML disponibles como herramientas para el modelado de sistemas de información. Con esta información deben construir un instrumento que permita de asociar conceptos con una estructura similar a la que se presenta a continuación:



Se deben asociar diseñar de tal manera que permita asociar el nombre del diagrama, con su definición y su tipo. Al terminar el instrumento, debe aplicarse a otro compañero y evaluar sus conocimientos en el tema.

Finalmente, el aprendiz deberá elegir uno de los diagramas UML consultados y aplicarlo a su proyecto formativo, debe ser diferente a los trabajados anteriormente (DFD, casos de uso, clases).

Esta actividad corresponde a una evidencia de desempeño y producto, las dos entregas (instrumento de asociación de conceptos y diagrama UML del proyecto) deberán subirse en el enlace de la plataforma Blackboard habilitado por el instructor.

**3.4 Actividades de transferencia:**

* **Descripción de las actividades:** Una vez realizadas las actividades de reflexión, contextualización y apropiación sobre los temas relacionados con la definición del contexto del software y sus herramientas de modelado, es el momento de socializar y realizar la transferencia de lo aprendido, por ende se plantea la siguiente acción.

**Acción 1: Reunión de analistas:**

Para esta actividad, los aprendices se reunirán en los grupos de proyecto. Posteriormente, deberán revisar el documento ERS IEEE 830 y retroalimentarse con los conocimientos adquiridos en esta guía. Para llevar a cabo la retroalimentación, se debe revisar la especificación de requerimientos, y anexar además, los diagramas de clases, casos de uso y prototipos no funcionales (interfaces de usuario) que permitan modelar la solución propuesta (realizados en una de las herramienta CASE trabajadas en las actividades anteriores).

Esta actividad corresponde a una evidencia de desempeño y producto, el documento ERS IEEE 830 completo (teniendo en cuenta las observaciones mencionadas anteriormente) deberá subirse en el enlace de la plataforma Blackboard habilitado por el instructor.

**4. ACTIVIDADES DE EVALUACIÓN**

|  |  |  |
| --- | --- | --- |
| **Evidencias de Aprendizaje** | **Criterios de Evaluación** | **Técnicas e Instrumentos de Evaluación** |
| **Evidencias de Conocimiento :**   * ***AP1\_ANALIZAR\_GA2\_AP3\_DIAGRAMA\_CLASES***   **Evidencias de Desempeño**   * ***AP1\_ANALIZAR\_GA2\_AP1\_DIAGRAMA\_FLUJO\_DATOS*** * ***AP1\_ANALIZAR\_GA2\_AT1\_IEEE\_830\_PARTE\_2***   **Evidencias de Producto**   * *AP1\_ANALIZAR\_GA2\_AP1\_DIAGRAMA\_FLUJO\_DATOS* * ***AP1\_ANALIZAR\_GA2\_AP2\_ACERCAMIENTO\_SCRUM*** * *AP1\_ANALIZAR\_GA2\_AP3\_DIAGRAMA\_CLASES* * *AP1\_ESPECIFICAR\_GA2\_AT1\_IEEE\_830\_PARTE\_1* | * (**DESEMPEÑO, PRODUCTO**)   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.   * (**CONOCIMIENTO, DESEMPEÑO**)   ELABORA LOS DIAGRAMAS UML, DE ACUERDO CON LAS CARACTERÍSTICAS DE CADA UNO DE ELLOS, BASADO EN LOS REQUERIMIENTOS DEL CLIENTE, UTILIZANDO HERRAMIENTAS CASE. | *Técnica didáctica activa*   * *La pregunta* * *Observación* * *Simulación* * *Estudios de caso* * *Entrevista* * *Valoración de productos* * *Pruebas*   *Instrumentos*   * *Ficha técnica* * *Portafolio* * *Escalas de observación* * *Listas de control* * *Registro anecdótico* * *Cuaderno* * *Cuestionarios (conocimiento) –Téc. pregunta* * *Lista de chequeo (desempeño) – Téc. Observación* * *Lista de cotejo – variables/indicadores (producto) – Téc. Comparación/mapa conceptual* * *Mapa conceptual* * *Resolución de problemas explicitando los pasos seguidos* * *Fichas de recogida de información* * *Trabajos monográficos, pequeñas investigaciones* * *Preguntas de analogías/diferencias* * *Preguntas de interpretación y/o elaboración de gráficos, mapas, estadísticas* * *Exámenes tradicionales, en todas sus variantes orales y escritos* * *Rúbrica* * *Pruebas de ensayo y composición* * *Preguntas de respuesta corta* * *Preguntas de texto incompleto* * *Preguntas de correspondencia o emparejamiento* * *Preguntas de opción múltiple* * *Preguntas de falso-verdadero (justificadas)* |

**5. GLOSARIO DE TÉRMINOS**

**Ciclo de vida del software:** El término ciclo de vida del software describe el desarrollo de software, desde la fase inicial hasta la fase final.

**UML:** UML son las siglas de “Unified Modeling Language” o “Lenguaje Unificado de Modelado”. Se trata de un estándar que se ha adoptado a nivel internacional por numerosos organismos y empresas para crear esquemas, diagramas y documentación relativa a los desarrollos de software (programas informáticos).

**Diagrama de caso de uso:** Representa una funcionalidad particular de un sistema. Se crea para ilustrar cómo se relacionan las funcionalidades con sus controladores (actores) internos/externos.

**Diagrama de clases:** El diagrama UML más comúnmente usado, y la base principal de toda solución orientada a objetos. Las clases dentro de un sistema, atributos y operaciones, y la relación entre cada clase. Las clases se agrupan para crear diagramas de clases al crear diagramas de sistemas grandes.

**6. REFERENTES BIBLIOGRÁFICOS**

# Algoritmos a fondo : con implementaciones en C y Java, Pablo Augusto Sznajdleder, 2012

Disponible en Antioquia/Complejo Norte General (005.3 S998a ) y otras ubicaciones

* Lógica de programación y algoritmos : libro guía, Silvina Caro Pineda, 2003

Disponible Antioquia/Complejo Norte General (005.1 C292l ) y otras ubicaciones

* Joyanes, Aguilar, Luis, et al. Fundamentos de programación: libro de problemas. Algoritmos, estructuras de datos y objetos (2a. ed.), McGraw-Hill España, 2003. ProQuest Ebook Central, <https://ebookcentral-proquest-com.bdigital.sena.edu.co/lib/senavirtualsp/detail.action?docID=3195276>.
* Mancilla, Herrera, Alfonso. Diseño y construcción de algoritmos, Universidad del Norte, 2014. ProQuest Ebook Central, <https://ebookcentral-proquest-com.bdigital.sena.edu.co/lib/senavirtualsp/detail.action?docID=4183551>.
* Algoritmia - Técnicas fundamentales de programación, Recuperado de Bases de datos SENA ENI: <http://www.eni-training.com.bdigital.sena.edu.co/client_net/mediabook.aspx?idR=273360>

**7. CONTROL DEL DOCUMENTO**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Nombre** | **Cargo** | **Dependencia** | **Fecha** |
| **Autor (es)** | LEE JARED ESCOBAR  ALEJANDRO MEJIA  JOHNATTAN JARAMILLO  JOSÉ DAVID VÉLEZ  AHARÓN AGUAS  ANDRES CUELLAR  EDILFREDO PINEDA  EDWY PATIÑO | INSTRUCTOR  INSTRUCTOR  INSTRUCTOR  INSTRUCTOR  INSTRUCTOR  INSTRUCTOR  INSTRUCTOR  INSTRUCTOR | CTGI TIC  CTGI TIC  CTGI TIC  CTGI TIC  CTGI TIC  CTGI TIC  CTGI TIC  CTGI TIC | 12/07/2018 |

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

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