**PROCESO DIRECCIÓN DE FORMACIÓN PROFESIONAL INTEGRAL**

**FORMATO GUÍA DE APRENDIZAJE**

**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 (si es formación Titulada): DESARROLLO E IMPLEMENTACIÓN DE SOFTWARE DE ACUERDO LAS NECESIDADES DEL SECTOR TEXTIL EN ANTIOQUIA
* Fase del Proyecto (si es formación Titulada): PLANEAR
* Actividad de Proyecto (si es formación Titulada): 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:
* CONSTRUIR EL MODELO CONCEPTUAL DEL MACROSISTEMA FRENTE A LOS REQUERIMIENTOS DEL CLIENTE, MEDIANTE EL USO E INTERPRETACIÓN DE LA INFORMACIÓN LEVANTADA, REPRESENTADO EN DIAGRAMAS DE CLASE, DE INTERACCIÓN, COLABORACIÓN Y CONTRATOS DE OPERACIÓN, DE ACUERDO CON LAS DIFERENTES SECUENCIAS, FASES Y PROCEDIMIENTOS DEL SISTEMA.
* VALORAR LA INCIDENCIA DE LOS DATOS EN LOS PROCESOS DEL MACROSISTEMA, TOMANDO COMO REFERENTE EL DICCIONARIO DE DATOS Y LAS MINIESPECIFICACIONES, PARA LA CONSOLIDACIÓN DE LOS DATOS QUE INTERVIENEN, DE ACUERDO CON PARÁMETROS ESTABLECIDOS.
* 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.
* INTERPRETAR EL INFORME DE REQUERIMIENTOS, PARA DETERMINAR LAS NECESIDADES TECNOLÓGICAS EN EL MANEJO DE LA INFORMACIÓN, DE ACUERDO CON LAS NORMAS Y PROTOCOLOS ESTABLECIDOS.
* ELABORAR EL INFORME DE LOS RESULTADOS DEL ANÁLISIS DEL SISTEMA DE INFORMACIÓN, DE ACUERDO CON LOS REQUERIMIENTOS DEL CLIENTE SEGÚN NORMAS Y PROTOCOLOS ESTABLECIDOS.
* Duración de la Guía: 400 horas.

**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 los participantes del proceso de formación.

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

* **Descripción de la(s) actividad(es):** Las siguientes actividades están diseñadas para permitirle al aprendiz plantear la solución a situaciones relacionadas con la construcción del software de acuerdo con los requerimientos del cliente, en los lenguajes de programación definidos en formación.
* **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. Actividad 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 por medio de una videollamada, se realizará una discusión con los aprendices referentes a las preguntas propuestas.

**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 funcional + UML

Los resultados esperados pueden ser como estos:





Para esta actividad, cada aprendiz 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

Estas preguntas se socializarán en una sesión virtual con el instructor.

**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
* Reglas y consejos para el DFD
* Niveles y capas del DFD: de los diagramas de contexto al pseudocódigo
* Ejemplos de cómo se pueden usar los DFD
* 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 socializar y resumir los aspectos más importantes del tema asignado en un mapa mental, el cual se explicará en videollamada. 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 y conceptos de los diferentes lenguajes de programación de un software o proyecto informático y la construcción de una solución informática, el aprendiz debe desarrollar las siguientes acciones:

**Acción 1: Introducción a Scrum.**

Los marcos de trabajo para el 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. [1]

En esta actividad, los aprendices tendrán su primer acercamiento a una de los frameworks de desarrollo ágil más populares en la industria: SCRUM, en la ilustración podrán ver las empresas que usan este framework de acuerdo con el sector económico.



*Ilustración 1. Empresas que usan SCRUM. (extraído de* [*https://proyectosagiles.org/historia-de-scrum/*](https://proyectosagiles.org/historia-de-scrum/)*)*

Para esta acción los aprendices deben realizar la lectura de las premisas y valores del SCRUM en las páginas 22 a la 25 del libro *Implantar scrum con éxito (*<https://ebookcentral-proquest-com.bdigital.sena.edu.co/lib/senavirtualsp/detail.action?docID=4795177> deben ingresar su documento de identidad en usuario y contraseña para ingresar al repositorio de la biblioteca del SENA).

Después de realizar la lectura de las páginas mencionadas del libro, deben consultar los doce principios del manifiesto ágil (<https://agilemanifesto.org/iso/es/principles.html>). Para complementar las lecturas y la información se recomienda que el aprendiz vea el siguiente video: <https://www.youtube.com/watch?v=HhC75IonpOU>

Luego se realizará una sesión en línea con el instructor donde se realizará una discusión de los conceptos aprendidos por los aprendices y una explicación general del ciclo scrum.

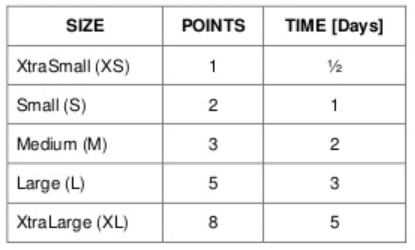
Adicionalmente, es importante conocer los roles y sus funciones, invitamos a ver el siguiente link.

<https://www.youtube.com/watch?v=kS9nEm-xUlM><https://youtu.be/8Wz7qDfGf8Q>

**Acción 2: Historias de usuario (user story), primer insumo UML.**

Una user story es una ficha que contiene toda la información relevante disponible sobre una funcionalidad del proyecto, se puede representar como una necesidad del cliente o un requerimiento funcional. Es la pieza fundamental del SCRUM y se definen según la regla INVEST:

* **Independent**: una historia debería ser independiente, de forma que permita moverla sin que esto afecte en exceso al resto de las historias.
* **Negotiable**: o, mejor dicho, adaptable. Aquello escrito en la historia puede cambiar a medida que el proyecto avanza.
* **Valuable**: o, mejor dicho, funcional. La historia tiene que describir un elemento de valor para el negocio.
* **Estimatable**: el equipo de desarrollo debe tener bastante información, y esta tiene que ser bastante clara para permitir una estimación de esfuerzo.
* **Sized appropriately:** la historia no tiene que ser ni demasiado grande, que impida que quepa en un sprint o que se pueda combinar con otras historias, ni demasiado pequeña, que no aporte valor por sí misma. Se pueden estimar con unos puntos de historia de acuerdo con la información de la ilustración 2.



***Ilustración 2****. Guía para asignar puntos de historia.*

* **Testable:** todas las historias de usuario tienen que aportar unos criterios de aceptación que, al final, serán los que los usuarios ejecuten para dar o no aceptación a la historia. [1].

Para esta acción se recomienda que los aprendices profundicen su conocimiento en las historias de usuarios en los siguientes videos:

* Scrum | Historias de Usuario: <https://www.youtube.com/watch?v=FJuq_lrM5Cc>
* ¿Cómo crear historias de usuario en scrum?: <https://www.youtube.com/watch?v=ky6wFiF5vMk>

Con la información consultada, los aprendices deben elaborar por grupos de proyectos las historias de usuarios (Mínimo cada proyecto debe tener 15 historias de usuario), de acuerdo con el formato de Excel alojado en la plataforma Territorium, este archivo se debe tener diligenciado de manera completa ya que se usará más adelante para la evidencia de desempeño y producto.

**3.4 Actividades de transferencia:**

* **Descripción de las actividades:** Una vez realizadas las actividades de reflexión, contextualización y apropiación sobre los conceptos básicos la construcción de una solución de software es el momento de construir una solución real aplicada al proyecto formativo, para ello se plantea la siguiente acción:

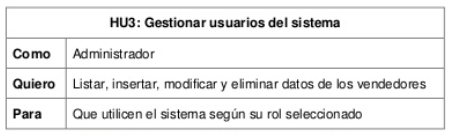
**Acción : Diagrama de casos de uso, una mirada al negocio**

Cuando se habla de diagrama se entiende por gráficos que se presentan en esquemas los cuales proporcionan información relativa e inherente a algún tipo de ámbito, ya sea político o económico, con este concepto básico sobre diagrama le invitó a trabajar la siguiente actividad. [2]

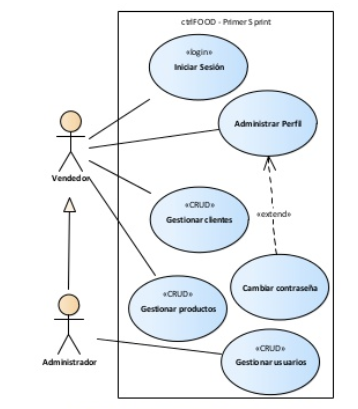
El instructor mediante una sesión en línea realizará la introducción a los casos de uso y su importancia en la etapa de análisis de requerimientos en los sistemas de información. Se revisarán aspectos como su definición, uso, componentes y ejemplos.

Cada aprendiz se documenta sobre los casos de uso, leyendo, Kimmel, P. (2008). *Manual de uml*. Retrieved from [https://ebookcentral-proquest-com.bdigital.sena.edu.co](https://ebookcentral-proquest-com.bdigital.sena.edu.co/), páginas 21 al 29.

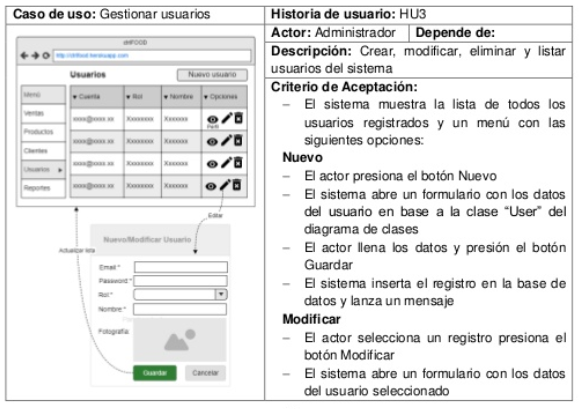
Luego de esto, los aprendices de manera individual observa el siguiente video. Curso UML: Diagrama Casos de Uso<https://www.youtube.com/watch?v=EbLkWdh1Hpk>. Posteriormente se reunirán en los equipos de trabajo, para realizar los diagramas de casos de uso aplicados a los requerimientos funcionales de su proyecto y relacionado a las historias de usuario que desarrollaron. Deben hacerlo siguiendo el ejemplo y el formato mostrado en las ilustraciones 3, 4 y 5.



***Ilustración 3****. Ejemplo de historia de usuario HU3 asociado a un inicio de sesión como administrador y como vendedor.*



***Ilustración 4.*** *Ejemplo de caso de uso de un sistema de inicio de sesión como administrador y como vendedor.*

**

***Ilustración 5.*** *Ejemplo de un caso de uso asociado a la historia de usuario HU3*.

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

Estos casos de uso se debe tener diligenciado de manera completa ya que se usará más adelante para la evidencia de desempeño y producto.

**NOTA:** No necesariamente se debe hacer un caso de uso por historia de usuario, puede ocurrir que un mismo caso de uso involucre varias historias de usuarios.

**Acción 2: Diagrama de actividades, flujograma del proceso.**

Para esta acción los aprendices deben realizar la lectura de la información en este enlace: <https://www.lucidchart.com/pages/es/tutorial-diagrama-de-actividades-uml>. Luego por medio de una sesión en línea con el instructor se solucionarán dudas y se mostrarán más ejemplos.

Luego de esta socialización los aprendices, distribuidos en sus grupos de trabajo desarrollarán un diagrama de actividades para cada caso de uso desarrollado en la actividad anterior. Estos diagramas deben tener diligenciado de manera completa ya que se usará más adelante para la evidencia de desempeño y producto.

**Acción 3: Diagramas de clase.**

Para esta acción los aprendices deben realizar la lectura de la información en este enlace: <https://www.lucidchart.com/pages/es/tutorial-de-diagrama-de-clases-uml>. Y ver el siguiente video: <https://www.youtube.com/watch?v=Z0yLerU0g-Q>

Luego de realizar la lectura y visualizar el video, se hará una solución de dudas por parte del instructor, por medio de una sesión en línea.

Para esta acción los aprendices se reunirán en grupos de proyectos, **en caso de que el número de grupos sea par** el instructor los enumerará y se formarán parejas de grupos entre el primero y el último, el segundo y el penúltimo, y así sucesivamente hasta completar, **en caso de que el número de grupos sea impar,**  el instructor decidirá que grupo trabajará solo y los otros se agruparán de acuerdo al procedimiento descrito anteriormente.

Cada grupo entregará a su grupo emparejado los casos de uso realizados en una acción anterior y de cada uno de estos caso de uso deben elaborar un diagrama de clases, posterior a esto cada grupo, sin ver el diagrama que sus compañeros hicieron de sus casos de uso, deberá realizar los diagramas de clase de sus propios casos de uso.

Entonces al finalizar la acción cada grupo deberá tener los diagramas de clase de cada uno de sus casos de uso hecho por ellos mismos y la realizada por sus compañeros, y con esta información harán una paralelo y definirán un diagrama de clases de su proyecto formativo.

Esta actividad se socializará con sus compañeros y el instructor mediante una sesión en línea.

Para el diseño del diagrama de clases, se sugiere el uso de herramientas CASE como StarUML, Dia, ArgoUML o Lucidchart.

**Acción 4: Ingreso a las Bases de datos.**

El aprendiz realizará una investigación de que es una BD, tipos de BD, que es una entidad, atributos, tipos de datos, Cardinalidades, normalización las 3FN, que es un Modelo Entidad Relacion - Modelo Relacional y un diccionario de datos.

**Profundizar en este enlace:**

[**https://www.youtube.com/watch?v=fVWXwKrmEmY&list=PLAzlSdU-KYwWodXmd8PeJY6zpv0wEwY0J**](https://www.youtube.com/watch?v=fVWXwKrmEmY&list=PLAzlSdU-KYwWodXmd8PeJY6zpv0wEwY0J)

**Formato - Diccionario de datos:**

[**https://drive.google.com/file/d/1DlIuMJxb6Ss-6w98tzlLsJR8-WbkwayK/view?usp=sharing**](https://drive.google.com/file/d/1DlIuMJxb6Ss-6w98tzlLsJR8-WbkwayK/view?usp=sharing)

El instructor le compartirá el formato para el diccionario de datos e impartirá una clase magistral para resolver dudas y como realizar los modelos (DD, MER, MR) de la BD.

El equipo de trabajo debe de entregar los modelados de la BD (diccionario de datos, MER, MR), estos modelos deben de estar apuntando a los requisitos del cliente de cada Proyecto Formativo.

**Acción 5. al RESCATE!!!**

La importancia de GIT:

Póngase en el lugar de una empresa de software donde muchos proyectos son desarrollados por muchos ingenieros y tecnólogos del SENA. Muchas líneas de código y todos aportando a ese mismo proyecto “MÁGNIFICO” todos trabajan sin problemas, esta tarea de sincronizar el código se hace de forma automática y lo hace GIT.

En el SENA por años los aprendices ADSI, intercambian los archivos por correos o usando usb “mándeme su código yo lo pego al proyecto”. Generalmente los proyectos se dañan, si se hacen cambios se pierde lo ultimo que se hace o se pierde la versión que funcionaba, o un compañero de grupo desertó y se no entregó el código del proyecto.

La salvación es GIT, aquí presentamos una guía rápida para aprender de forma fácil esta herramienta

<https://rogerdudler.github.io/git-guide/index.es.html>

Lo primero es crear una cuenta git en el sitio oficial, y luego sincronizar su repositorio local con el remoto.

En esta guía aprenderás hacer un commit, a crear una rama, a generar updates y hacer merges de sus proyectos.

La meta de esta actividad es subir via línea de comandos su proyecto a un repositorio remoto y compartir el proyecto con sus compañeros e instructores. En otro momento de aprendizaje habrá la segunda parte de git más avanzado.

**Recuerden que todo debe quedar en el repositorio de git hub, compartido con los instructores:**

**CÓDIGO, DIAGRAMAS, PRESENTACIÓN POWERPOINT.**

**Acción 6. ALGORITMOS SECUENCIALES, CONDICIONALES, CICLOS, VECTORES Y MATRICES.**

Una metodología es básicamente un conjunto de estrategias a fines que se ejecutan paso a paso para lograr el producto final o programa. Las metodologías en desarrollo de software también se conocen bajo el nombre de paradigmas los cuales son seleccionados de acuerdo a la naturaleza del proyecto a desarrollar.

En el análisis definirán el dominio del problema definiendo los datos de entrada y salida requeridos por los diferentes procesos implicados en los algoritmos a desarrollar. Se diseñará el correspondiente seudocódigo y DFD (Diagramas de Flujo de Datos). Recomendado: Aguilar, L. (2008). Fundamentos de programación. (4a. ed.) CAPÍTULO 2 Metodología de la programación y desarrollo de software

**Acción 6.1. Descripción pasó a paso de un evento.**

El instructor impartirá una clase magistral relacionada con los algoritmos y lógica de programación.

Cada aprendiz orientado por el instructor, explicará el paso a paso de un modelo, como, por ejemplo: secuencia de pasos lógicos para cambiar la llanta de un automóvil.

**SECUENCIA LÓGICA**

* Colocar la llanta buena
* Orillar el carro
* Reanudar el viaje
* Detener el carro
* Abrir la maleta del carro
* Guardar la llanta pinchada, la herramienta y las señales
* Sacar las señales de peligro y colocarlas a cierta distancia
* Sacar la llanta y herramientas de la maleta del carro
* Introducir la llave en la tuerca
* Aflojar levemente las tuercas
* Levantar mediante un gato hidráulico el carro
* Sacar las tuercas
* Sacar la llanta pinchada
* Bajarse del carro
* Apretar las tuercas
* Subirse al carro

**Acción 6.2. Algoritmos.**

En cada sesión el instructor entregará misceláneas de algoritmos a los aprendices con el fin de ir afianzando sus conocimientos.

Ejercicios algoritmos1.

Ejercicios algoritmos2.

Ejercicios algoritmos3.

Ejercicios algoritmos4.

Ejercicios algoritmos5.

Ejercicios algoritmos6.

Ejercicios algoritmos7.

Ejercicios algoritmos8.

**Acción 7.** ALGORITMOS PARA SU PROYECTO que significa lo amarillo (Que hace falta)

Para la siguiente actividad debes tener claro los conocimientos anteriores los tipos de datos y estructuras de control anteriormente visto en las acciones anteriores, también debes tener claro los requisitos de su proyecto e historias de usuario. El siguiente reto es Aplicando la lógica desarrollar 3 algoritmos aplicados a su proyecto, donde el equipo de trabajo colaborativo identifiquen cuáles son los actores y procesos donde se debe tener mayor cuidado con la lógica, con las validaciones. Los algoritmos deben estar codificados en un lenguaje de programación (PHP, JAVA, C#, PYTHON). Si es posible el grupo puede entregar una interfaz en consola o gráfica donde se refleje el funcionamiento de los algoritmos, pidiendo los datos de entrada y mostrando resultados de dichos algoritmos. Esta acción se socializa en una sesión sincrónica con un instructor.

En experiencias pasadas, se han generado algoritmos como la ruta más cercana (algoritmo de dijkstra), o algoritmos para reservar las sillas de una sala de cines, algoritmos para totalizar una factura, el reto final es que ustedes consideren donde van a tener mayor exigencia de lógica para que planteen un buen algoritmo.

**Acción 8.** **SPRING ENTREGA DE FASE**

El aprendiz debe preparar una entrega de FASE con su equipo, con esta nueva metodología de SCRUM la vamos a llamar spring review. Debes tener claro diversos conceptos de la metodología SCRUM, roles, y procesos que se aplican en esta metodología.

Los entregables son:

1. diagramas UML.

2. Diseño de la base de datos

3. Presentación power point, problema, justificación, objetivos general y específicos (inglés)

4. Para esta entrega documentarse de cómo se hace un **spring review**

**clic aquí->** [**Scrum Ejemplo Práctico Revisión del Sprint**](https://www.youtube.com/watch?v=GDu_pcXCXus)

Para esta entrega, se debe tomar atenta nota de las observaciones del scrum master, de los stakeholders y de los product owners.

Pueden invitar a esta entrega personas o entidades externas que les interese el proyecto y que hagan las veces de stakeholders.

Después de realizar el spring review el equipo de trabajo debe realizar dos reuniones adicionales las cuales serán las siguientes:

Las salidas de esta actividad son las siguientes:

1. Generar back log totalmente priorizado.

2. Desarrollar un spring retroespective

En el siguiente link, nos explica de forma clara y detallada como y que se se hace en el spring retroespective, elementos como seguir haciendo, hacer más, empezar a hacer, dejar de hacer es muy importante trabajarlos aquí.

clic aquí> [Reunión retrospectiva](https://www.youtube.com/watch?v=7wd61RrJAfo)

Es importante realizar una retrospectiva “A intervalos regulares el equipo reflexiona sobre cómo ser más efectivo para a continuación ajustar y perfeccionar su comportamiento en consecuencia”



***Ilustración 7.*** *Spring review estrella de mar*.

**Recuerden que todo debe quedar en el repositorio de git hub, compartido con los instructores:**

**CÓDIGO, DIAGRAMAS, PRESENTACIÓN POWERPOINT.**

ENTREGABLES DE LA ACCIÓN EN TERRITORIUM:

1. entrega de back log de su proyecto.
2. entrega de un documento con las lecciones más pertinentes de la retrospectiva

**Evaluación: El equipo de instructores gestiona la lista de chequeo con los ítems que se van a evaluar.**

**4. ACTIVIDADES DE EVALUACIÓN**

Tome como referencia la técnica e instrumentos de evaluación citados en la guía de Desarrollo Curricular

|  |  |  |
| --- | --- | --- |
| **Evidencias de Aprendizaje** | **Criterios de Evaluación** | **Técnicas e Instrumentos de Evaluación** |
| **Evidencias de Conocimiento :**  check list de conceptos y principios de scrum  Prueba de conocimiento.  **Evidencias de Desempeño y Producto:**  **DIAGRAMAS UML.**  Realiza la entrega del diagrama de **Casos de Usos** de forma coherente cumpliendo con los requerimientos del cliente.  Realiza la entrega del Diagrama de **Clases** de forma coherente cumpliendo con los requerimientos del cliente.  Realiza la entrega del Diagrama de **Actividades** de forma coherente cumpliendo con los requerimientos del cliente.  Realiza la entrega de la **Historia de Usuario** de forma coherente cumpliendo con los requerimientos del cliente.  **BASES DE DATOS.**  Realiza la entrega del **Diccionario de Datos** de forma Coherente cumpliendo con los requerimientos del cliente.  Realiza la entrega del **Modelado MER (MODELO ENTIDAD RELACION)** de forma Coherente cumpliendo con los requerimientos del cliente.  Realiza la entrega del **Modelado MR (MODELO RELACIONAL)** de forma Coherente cumpliendo con los requerimientos del cliente.  **ALGORITMIA.**  Realiza la entrega de los **Algoritmos Secuenciales.**  Realiza la entrega de los **Algoritmos con Condicionales.**  Realiza la entrega de los **Algoritmos con Ciclos.**  Realiza la entrega de los algoritmos en los diferentes **Lenguajes de programación (Python, Java, Visual Studio, PHP).** | * 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. * 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. * 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. * 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. * Realiza el informe de análisis del sistema de información requerido, presentando el estado actual de la organización, y proponiendo alternativas de solución, de acuerdo con las normas y protocolos establecidos. | * **Técnica didáctica activa**   Observación   * I**nstrumentos**   Lista de verificación. |

**5. GLOSARIO DE TÉRMINOS**

**6. REFERENTES BIBLIOGRÁFICOS**

[1] Monte, G. J. (2016). *Implantar scrum con éxito*. Retrieved from <https://ebookcentral-proquest-com.bdigital.sena.edu.co>

[2]Kimmel, P. (2008). *Manual de uml*. Retrieved from <https://ebookcentral-proquest-com.bdigital.sena.edu.co>

Aguilar, L. (2008). Fundamentos de programación. (4a. ed.) McGraw-Hill Interamericana. Tomado de http://www.ebooks7-24.com.bdigital.sena.edu.co/?il=524

**BIBLIOGRAFÍA RECOMENDADA**

Monte, G. J. (2016). *Implantar scrum con éxito*. Retrieved from <https://ebookcentral-proquest-com.bdigital.sena.edu.co>

las, H. D. D. R. D., & Álvarez, G. A. (2018). *Métodos ágiles : Scrum, kanban, lean*. Retrieved from <https://ebookcentral-proquest-com.bdigital.sena.edu.co>

Kimmel, P. (2008). *Manual de uml*. Retrieved from https://ebookcentral-proquest-com.bdigital.sena.edu.co

Oviedo, E. (2015). Lógica de programación orientada a objetos. Ecoe Ediciones. Tomado de http://www.ebooks7-24.com.bdigital.sena.edu.co/?il=1034

**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 | **31/01/2019** |

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

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Nombre** | **Cargo** | **Dependencia** | **Fecha** | **Razón del Cambio** |
| **Autor (es)** | LEE JARED ESCOBAR  ALEJANDRO MEJIA  AHARÓN AGUAS  EDILFREDO PINEDA  EDWY PATIÑO | INSTRUCTORES | CTGI TIC | 14/10/2020 | Actualización formato y anexo de contenido de estadística para la tabulación y elaboración de gráficas de los resultados de encuestas. |