



PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL
FORMATO GUÍA DE APRENDIZAJE

IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE

- Denominación del Programa de Formación: **Técnico en Programación de Software**
- Código del Programa de Formación: **233104 v.1**
- Nombre del Proyecto: **Desarrollo de un software de acuerdo con la interpretación de los requerimientos aplicando las competencias del técnico en programación de software**
- Fase del Proyecto: **Fase II - Interpretación**
- Actividad de Proyecto: **Interpretar la información técnica de diseño - Interpretar problemas ambientales, interpretar sentido de comunicación comprender Información básica en ingles**
- Competencia: **Desarrollar la solución de software de acuerdo con el diseño y metodologías de desarrollo.**
- Resultados de Aprendizaje Alcanzar: **Interpretar la información técnica de diseño para la codificación del software.**
- Duración de la Guía: **70 horas**

2. PRESENTACIÓN



Actualmente el ser humano se enfrenta al desarrollo de soluciones del entorno, esto conlleva a tener un aprendizaje previo, de retroalimentación, además de que este aprendizaje se deriva en ocasiones de acciones y actividades que han concluido en fracasos, pero con la experiencia, el aprendizaje y conocimientos oportunos, se logra contar con procesos, diseños y procedimientos, que generan la solución de manera satisfactoria.

Dentro de este proceso de formación integral, la presente guía busca orientar en el aprendizaje colaborativo y de conocimiento, que permita garantizar la interpretación de los requisitos en la etapa previa al desarrollo del software, garantizando propuestas de solución eficientes y acordes al conocimiento adquirido.

Los invitamos para que con su esfuerzo y dedicación, logremos el desarrollo en este proceso de formación y nuevos aprendizajes, que permitan el fortalecimiento de los conocimientos técnicos en el desarrollo de software.



3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

3.1 Actividad de reflexión inicial

Reflexionar sobre la importancia que tiene la interpretación de requerimientos de un software, identificar el análisis, las características, importancia y conceptos de los requerimientos en un software.

El instructor dispone del ambiente de aprendizaje para la actividad de formación, inicia con una bienvenida a los aprendices y presenta el video sobre análisis y requerimientos para un software.

Video: <https://www.youtube.com/watch?v=50aSAUFrALk>

Después de ver el video el instructor pasará a realizar un conversatorio con los aprendices (presencial o digital), sobre los conceptos desarrollados en el video, los aprendices brindarán sus aportes con referencia al aprendizaje adquirido, para el afianzamiento de los conceptos desarrollados.

Duración: 2 horas

Ambiente requerido: Ambiente de aprendizaje con Equipos de Cómputo y Software Especializado.

Materiales: Marcadores de diferente color, Video beam o TV, bafles, Conexión a Internet

3.2 Actividades de contextualización e identificación de conocimientos



Identificar las generalidades sobre los conceptos de programación de software, Identificar los diferentes conceptos y términos que se aplican a la programación de software.

En el ambiente de aprendizaje, el instructor entregará un cuestionario en forma virtual en la plataforma educativa, para que los aprendices identifiquen los conocimientos que poseen de los temas: informe técnico de requerimientos del sistema, requerimientos funcionales, requerimientos no funcionales, análisis de requerimientos, diagramas de casos de uso, diagramas de clases, diagramas de objetos.

Para este proceso de consulta, los aprendices tendrán un tiempo de 1 hora, el documento lo subirán como evidencia de aprendizaje de conocimiento.

Después de este proceso el instructor realizará una mesa redonda en un tiempo de 1 hora, con los aprendices para afianzar los conceptos de desarrollo de software consultados.

Evidencia de aprendizaje de conocimiento: Taller de consulta sobre conceptos de desarrollo de software

Duración: 2 horas

Ambiente requerido: Ambiente de aprendizaje con Equipos de Cómputo y Software Especializado.

Materiales: Marcadores de diferente color, Video beam o TV, bafles, tripode, Conexión a Internet



3.3 Actividad de apropiación del conocimiento (conceptualización y teorización)

3.3.1 Reconocer los conceptos de programación orientada a objetos



COGNITIVA: Analizar e interpretar el informe técnico de diseño, para realizar la construcción del software.

PROCEDIMENTAL: Desarrollar el software con las especificaciones del informe técnico.

ACTITUDINAL: Demostrar Interés por organizar y fundamentar las especificaciones del informe en el desarrollo del software.

En el ambiente de aprendizaje el instructor iniciara con la presentación de los siguientes dos videos:

Video 1: <https://www.youtube.com/watch?v=AonGYo7VV7Y>

Video 2: <https://www.youtube.com/watch?v=ykpT5P7171M>

Despues de ver los videos para afianzar los conocimientos, los aprendices realizaran la lectura del documento de programación orientada a objetos.

Lectura: <http://luis.izqui.org/resources/ProgOrientadaObjetos.pdf>

El instructor realizara un feedback sobre programación orientada a objetos, concepto y ejemplos de clases, objetos e instancia, incluyendo además los fundamentos de la programación orientada a objetos (encapsulación, abstracción, herencia y polimorfismo)

Los aprendices realizaran un estudio de casos de programación orientada a objetos y presentaran su respectivo informe.

Duración: 10 horas

Ambiente requerido: Ambiente de aprendizaje con Equipos de Cómputo y Software Especializado.

Materiales: Marcadores de diferente color, Video beam o TV, bafles, Conexión a Internet

3.3.2 Identificar las metodologías de desarrollo agiles de software



Identificar las características de las metodologías agiles y aplicación en el desarrollo de software

Concepto de metodologias agiles

Caracteristicas y ventajas de las metodologías agiles

Tipos de metodologías

XP (Extreme Programming)

SCRUM

Kanban

Agile Inception

Design Sprint, la metodología de Google

GFPI-F-135 V01

En grupos de 3 aprendices, realizaran una presentación y exposición de los temas propuestos de metodologías agiles.



Duración: 8 horas

Ambiente requerido: Ambiente de aprendizaje con Equipos de Cómputo y Software Especializado.
Materiales: Marcadores diferente color, Video beam o TV, bafles, Conexión a Internet

3.3.3 Reconocer los requisitos funcionales y no funcionales de un software



Reconocer los requisitos funcionales y no funcionales de un software para interpretar la información técnica de diseño

En el ambiente de aprendizaje el instructor realizara una presentación de los conceptos de requisitos funcionales y no funcionales que se tienen para un software. Las técnicas para identificar los requisitos y la identificación de la historia de usuario.

Los aprendices realizaran y presentaran un caso de estudio para afianzar el proceso de aprendizaje, donde identifiquen los requisitos funcionales y no funcionales.

Evidencia de aprendizaje de desempeño: Documento de verificación a traves de una lista de chequeo del software requerido de acuerdo al informe técnico.

Duración: 10 horas

Ambiente requerido: Ambiente de aprendizaje con Equipos de Cómputo y Software Especializado.
Materiales: Marcadores diferente color, Video beam o TV, bafles, Conexión a Internet

3.3.4 Reconocer los fundamentos de casos de uso en la interpretación técnica de requerimientos



El instructor realizara una presentación sobre los fundamentos de casos de uso: definición, representación gráfica, descripción, elementos (límite, escenario, actor, caso de uso, relaciones, asociación, dependencia o instanciación, generalización, herencia, include, extends), diagrama de clases y objetos.

El instructor utilizara un software o aplicación para creación de diagramas UML para realizar el modelamiento de los casos de uso, clases y objetos.

Despues los aprendices iniciaran a traves de un proceso de un aprendizaje basado en proyectos, la solucion de un caso de estudio aplicando los conocimientos adquiridos y presentaran los diagramas del respectivo caso.

Duración: 20 horas

Ambiente requerido: Ambiente de aprendizaje con Equipos de Cómputo y Software Especializado.

Materiales: Marcadores diferente color, Video beam o TV, bafles, Conexión a Internet



3.4 Actividades de transferencia



Construir el informe de requerimientos de software para evidenciar las actividades desarrolladas en el proceso formativo.

El instructor entregara a traves de la plataforma virtual, un caso de estudio sobre un informe técnico de diseño, para que los aprendices desarrollen los conocimientos adquiridos en un informe sobre los requerimientos que se tienen del software, desde el punto de vista técnico, argumentando los conceptos y temáticas que se deben abordar para su solución, incluyendo los diagramas de UML, requerimientos funcionales y no funcionales y los conceptos de fundamentación en programación de software.

Evidencia de desempeño de producto: Informe con los requerimientos necesarios para iniciar el proceso de construcción del software

Duración: 10 horas

Ambiente requerido: Ambiente de aprendizaje con Equipos de Cómputo y Software Especializado.
Materiales: Marcadores diferente color, Video beam o TV, bafles, Conexión a Internet

4. ACTIVIDADES DE EVALUACIÓN

Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
Evidencias de Conocimiento: Taller de consulta sobre conceptos de software	COMPRENDE EL INFORME TÉCNICO DE DISEÑO ENTREGADO.	Instrumento: Lista de chequeo o verificación
Evidencias de Desempeño: Documento de verificación a traves de una lista de chequeo del software requerido de acuerdo al informe técnico	DISTINGUE LOS ARTEFACTOS NECESARIOS PARA INICIAR EL PROCESO DE CONSTRUCCIÓN DEL SOFTWARE	Instrumento: Lista de chequeo o verificación
Evidencias de Producto: Informe con los requerimientos necesarios para iniciar el proceso de construcción del software.		Instrumento: Lista de chequeo o verificación



5. GLOSARIO DE TÉRMINOS

Modelo: es una representación de un concepto de software.

Desarrollador de software: es un programador o una empresa comercial que se dedica a uno o más aspectos del proceso de desarrollo de software

UML: Lenguaje Unificado de Modelado (UML) fue creado para forjar un lenguaje de modelado visual común y semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento

Programación orientada a objetos: La programación Orientada a objetos se define como un paradigma de la programación, una manera de programar específica, donde se organiza el código en unidades denominadas clases, de las cuales se crean objetos que se relacionan entre sí para conseguir los objetivos de las aplicaciones.

Algoritmo: es un conjunto de instrucciones o reglas definidas y no-ambiguas, ordenadas y finitas que permite, típicamente, solucionar un problema, realizar un cómputo, procesar datos y llevar a cabo otras tareas o actividades.

Lenguajes de programación: es un lenguaje con reglas gramaticales bien definidas, que le proporciona a una persona, en este caso el programador, la capacidad de escribir una serie de instrucciones o secuencias de órdenes en forma de algoritmos con el fin de controlar el comportamiento de un proceso físico o lógico de una computadora, de manera que se puedan obtener diversas clases de datos o ejecutar determinadas tareas.

Bases de datos: conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

6. REFERENTES BIBLIOGRÁFICOS

Pressman, R (2010). Ingeniería del software un enfoque práctico. Séptima edición. McGRAW-HILL INTERAMERICANA EDITORES, S.A.

Metodología Gestión de Requerimientos. Consultado en:
<https://sites.google.com/site/metodologiareq/capitulo-i>

UML. <https://ingenieriasoftware2011.files.wordpress.com/2011/07/el-lenguaje-unificado-de-modelado-manual-de-referencia.pdf>

Villena, J. Programación orientada a objetos. Consultado en: <http://www.it.uc3m.es/java/git-gisc/units/oo-herencia/slides/ProgramacionOrientadaAObjetos.pdf>

Juganaru, M. Introducción a la programación. Consultado en:
<https://www.editorialpatria.com.mx/pdf/files/9786074384154.pdf>



7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)		Instructor Instructor Instructor		2/12/2020

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

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					