PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL

GUÍA DE APRENDIZAJE N.º 04

1. IDENTIFICACIÓN DE LA GUIA DE APRENDIZAJE

Programa de Formación: ***Análisis y Desarrollo de Software***	**Código:** *228118* **Versión:** 1.0
Nombre del Proyecto: *Construcción de software integrador de tecnologías orientadas a servicios*	**Código:** 228118 v 1.0
Fase del proyecto: *4. Planeación*	**Ambiente de formación:** *Ambiente 2: Tecnologías web*
Actividad (es) del Proyecto: ***AP.1.*** *Diseñar la estructura de la base de datos del sistema* ***AP.2.*** *Optimizar el modelo de datos de acuerdo con estándares y políticas de seguridad* ***AP.3.*** *Determinar la plataforma tecnológica a utilizar para el desarrollo del software* ***AP.4.*** *Determinar estructura lógica del software a través del diagrama de clases y de paquetes* ***AP.5****. Elaborar los artefactos necesarios para el despliegue del software* ***AP.6.*** *Realizar la evaluación de artefactos del software*	
Resultados de Aprendizaje: ***RAE: 1.*** *Elaborar los artefactos de diseño del software siguiendo las prácticas de la metodología seleccionada.* ***RAE: 2.*** *Estructurar el modelo de datos del software de acuerdo con las especificaciones del análisis.* ***RAE: 3.*** *Determinar las características técnicas de la interfaz gráfica del software adoptando estándares.* ***RAE: 4.*** *Verificar los entregables de la fase de diseño del software de acuerdo con lo establecido en el informe de análisis.*	**Competencia:** ➢ *Modelado de los artefactos del software.*
Duración de la guía (en horas): *336 horas (42 sesiones)*	**Materiales de formación**: Consultar en cada actividad de aprendizaje.

2. PRESENTACION

El modelado de artefactos de software es una práctica fundamental en el desarrollo de software, ya que permite la creación de un diseño preciso y detallado del software a desarrollar. En esta etapa, se deben elaborar los artefactos de diseño del software siguiendo las prácticas de la metodología seleccionada, lo que incluye la definición de los requerimientos, la especificación de los componentes y la estructuración del modelo de datos.



RECONOCIMIENTO DE APRENDIZAJES NECESARIOS PARA EL APRENDIZAJE

En el proceso de alistamiento en su trabajo, usted debe asegurarse de que tiene todos los conocimientos y habilidades necesarias para llevar a cabo sin contratiempos la labor encomendada. Por favor responda el cuestionario para identificar las necesidades en sus aprendizajes junto con un diagnóstico donde uste valorará sus conocimientos previos. Para nada es malo responder "No conozco la respuesta", ya que en la primera clase se hará un repaso de estos temas.



El link del cuestionario es: https://forms.office.com/r/KU5g72iCXF
Después realimente en el siguiente marque con una X el rango de los resultados obtenidos y analice con su instructor los esfuerzos y dedicación necesaria para el aprendizaje que recién iniciará.

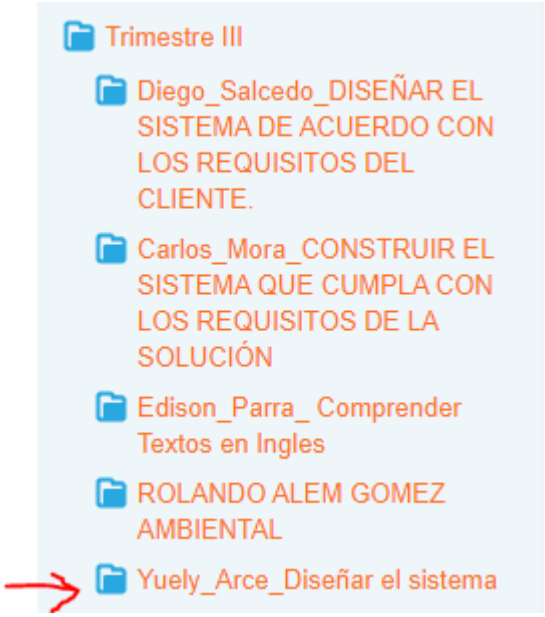
CONTEXTUALIZACION

La empresa ***TechVantage Solutions*** quiere renovar su sitio web y necesita un diseño moderno y atractivo que refleje su marca y atraiga a más clientes.

El problema a resolver es: ¿Cómo diseñar y desarrollar un sitio web moderno y atractivo que refleje la marca de ***TechVantage Solutions*** y aumente su número de clientes?

Para resolver este problema, se deben tomar en cuenta varios factores como:

- Analizar la marca de ***TechVantage Solutions***, sus valores y estilo para crear un diseño coherente con ellos.
- Identificar las necesidades y preferencias de los usuarios de ***TechVantage Solutions*** para diseñar un sitio web que les resulte atractivo y fácil de usar.
- Definir la estructura y organización del sitio web para facilitar la navegación del usuario y mejorar la experiencia del usuario.
- Utilizar tecnologías y herramientas adecuadas para desarrollar el sitio web, garantizando que sea rápido, seguro y adaptable a diferentes dispositivos y pantallas.



Al resolver este problema, la empresa de desarrollo de software podrá ofrecer a ***TechVantage Solutions*** un sitio web moderno y atractivo que refleje su marca y que le permita atraer a más clientes, mejorar su presencia en línea y aumentar sus ventas.

"Aprender sin pensar es inútil. Pensar sin aprender, peligroso". Confucio.

3. FORMULACION DE LAS ACTIVIDADES DE APRENDIZAJE

Es el momento de potenciar nuestras habilidades en especificación de propuesta de software para el desarrollo de proyectos de software. Esta guía requiere de una serie de actividades anexas las cuales se encuentran ubicadas como se indica en la figura. Recuerda que en la plataforma **LMS (Territorium) www.senavirtual.edu.co** se encuentra todo el material y los enlaces para la recolección de evidencias. Nunca olvidar que para aprobar la competencia relacionada se debe aprobar todas las evidencias aportadas por usted y que éstas a su vez también sean subidas a la plataforma.

Actividad 1. Diseñar la estructura de la base de datos del sistema

Grupos de trabajo: parejas

- *Tiempo de ejecución: 7,5 sesiones (60 horas)*

Objetivo: Diseñar la estructura de la base de datos del sistema para garantizar que la información se almacene de manera eficiente y se pueda acceder fácilmente a ella.

Pasos:

- Identificar los requisitos de información del sistema para saber que se debe almacenar en ella y cómo se utilizará. Esto se puede hacer en conjunto con los usuarios del sistema.
- Definir las entidades y relaciones con base en los requisitos de información identificados.
- Crear el diagrama entidad-relación (ER) que muestre visualmente la estructura de la base de datos y cómo se relacionan las entidades.
- Normalizar la base de datos para garantizar que la información se almacene de manera eficiente y se pueda acceder fácilmente a ella. Eliminando redundancias y estableciendo restricciones para garantizar que la información sea coherente y precisa.
- Crear el modelo relaciona que especifica cómo se deben estructurar las tablas y cómo se relacionan entre sí.
- Definir las restricciones y reglas de integridad que establezcan cómo se pueden ingresar, modificar y eliminar datos de la base de datos.
- Realizar pruebas y ajustes para asegurarse de que la base de datos funciona correctamente y ajustarla en caso de ser necesario.

Actividad 2. Optimizar el modelo de datos de acuerdo con estándares y políticas de seguridad

- *Grupos de trabajo: parejas*
- *Tiempo de ejecución: 7,5 sesiónes (60 horas)*

Objetivo: Optimizar el modelo de datos del sistema para cumplir con estándares y políticas de seguridad, garantizando que la información se almacene de manera segura y se proteja contra posibles ataques externos.
Pasos:

- Revisar los estándares y políticas de seguridad que aplican al sistema.
- Identificar las vulnerabilidades del modelo de datos actual y cómo podrían ser explotadas por atacantes externos.
- Aplicar técnicas de seguridad a la estructura de la base de datos con base en los estándares y políticas de seguridad y las vulnerabilidades identificadas.
- Optimizar el rendimiento de la base de dato para garantizar que el sistema funcione de manera eficiente y que la información se acceda rápidamente. Se deben aplicar técnicas como la indexación adecuada y la optimización de consultas.
- Verificar la integridad de los datos para garantizar que se almacenan de manera coherente y precisa. Se deben establecer restricciones y reglas de integridad para asegurarse de que los datos ingresados sean válidos y se almacenen de manera consistente.
- Realizar pruebas y ajustes en caso de ser necesario.

Actividad 3. Determinar la plataforma tecnológica a utilizar para el desarrollo del software

- *Grupos de trabajo: individual*
- *Tiempo de ejecución: 5 sesiones (40 horas)*

Objetivo: Seleccionar la plataforma tecnológica adecuada para el desarrollo del software, teniendo en cuenta las necesidades del proyecto y las características de las diferentes opciones disponibles.
Pasos:

- Identificar las necesidades del proyecto para determinar la plataforma tecnológica más adecuada. Esto se puede hacer a través de reuniones con los stakeholders, donde se identifiquen los requisitos y las características clave del software.
- Investigar las opciones de plataforma tecnológica disponibles. Esto puede incluir lenguajes de programación, frameworks, sistemas operativos, bases de datos, entre otros.
- Evaluar las opciones de plataforma tecnológica, teniendo en cuenta factores como la facilidad de uso, el costo, la escalabilidad, la seguridad y la disponibilidad de recursos y herramientas de soporte.
- Seleccionar la plataforma tecnológica adecuada más adecuada para el proyecto. Se deben considerar las necesidades del proyecto y las limitaciones presupuestarias y técnicas.
- Planificar la implementación de la plataforma tecnológica. Esto puede incluir la adquisición de herramientas y recursos necesarios, así como la capacitación del equipo de desarrollo en el uso de la plataforma.

Actividad 4. Determinar estructura lógica del software a través del diagrama de clases y de paquetes

Grupos de trabajo: individual

- *Tiempo de ejecución: 5 sesiones (40 horas)*

Objetivo: Diseñar la estructura lógica del software a través del diagrama de clases y de paquetes, lo que permitirá identificar y organizar las distintas clases y paquetes que conforman el sistema, y establecer las relaciones entre ellos.

Pasos:

- Identificar las clases del sistema, teniendo en cuenta las funcionalidades que se deben implementar y la relación entre ellas.
- Agrupar las clases en paquetes asegurando que deben estar organizados de manera lógica y coherente.
- Identificar las relaciones entre las clases, como la herencia, la composición o la asociación.
- Representar las clases y los paquetes en un diagrama de clases y de paquetes, que permita visualizar de manera clara y organizada la estructura lógica del software.
- Validar el diagrama de clases y de paquetes con los stakeholders y el equipo de desarrollo, para asegurarse de que la estructura lógica del software sea coherente y se ajuste a las necesidades del proyecto.

Actividad 5. Elaborar los artefactos necesarios para el despliegue del software

- *Grupos de trabajo: individual*
- *Tiempo de ejecución: 4,5 sesiones (36 horas)*

Objetivo: Elaborar los artefactos necesarios para el despliegue del software, lo que permitirá asegurar que el proceso de implementación se realice de manera correcta y eficiente.

Pasos:

- Identificar los requisitos del entorno de producción donde se va a desplegar el software, como la configuración del sistema operativo, las bibliotecas necesarias, el espacio en disco, entre otros.
- Definir el proceso de instalación del software, incluyendo la secuencia de pasos, los archivos necesarios, las dependencias y las configuraciones requeridas.
- Generar los paquetes de instalación del software, utilizando herramientas y técnicas que permitan crear paquetes completos y confiables, como instaladores y paquetes de actualización.
- Documentar los pasos de instalación en un manual de usuario, para que los usuarios puedan instalar y configurar el software de manera correcta.
- Realizar pruebas de instalación en un entorno de pruebas, para asegurarse que se realice de manera correcta y sin errores.

Actividad 6. Realizar la evaluación de artefactos del software

- *Grupos de trabajo: individual*
- *Tiempo de ejecución:5 sesiones (40 horas)*

Objetivo: Realizar la evaluación de los artefactos del software, como documentos de diseño, código fuente, pruebas y documentación del usuario, para asegurar que cumplan con los requisitos y estándares establecidos.

Pasos:

- Identificar los artefactos a evaluar, como documentos de diseño, código fuente, pruebas y documentación del usuario.
- Definir los criterios de evaluación para cada uno de los artefactos, teniendo en cuenta los requisitos y estándares establecidos para el proyecto.
- Realizar la evaluación de los artefactos, utilizando herramientas y técnicas que permitan identificar posibles errores o incumplimientos de los requisitos y estándares establecidos.
- Documentar los resultados de la evaluación en un informe, que permita identificar las áreas de mejora y las recomendaciones para corregir los errores o incumplimientos encontrados.
- Implementar las correcciones de los errores o incumplimientos encontrados en la evaluación, siguiendo los procedimientos establecidos.
- Realizar pruebas de validación para verificar que las correcciones implementadas cumplan con los requisitos y estándares establecidos.

4. ACTIVIDADES DE EVALUACIÓN

Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación	Resultados de aprendizaje
EL SER			

Evidencias de Desempeño: Cumplimiento de reglamento del aprendiz, desempeño disciplinario, desempeño académico, solución de problemas, relaciones interpersonales, organización, trabajo en equipo, ambiental.	Actitudes y valores frente a diferentes aspectos de la cotidianidad según los estándares laborales y sociales.	**Técnica:** ● Observación directa **Instrumento** ● Lista de Chequeo: **Evidencia D_04-00**	
EL SABER **Evidencia de Conocimiento:** **Actividad 1.** • Identificar los requisitos de información del sistema para saber que se debe almacenar en ella y cómo se utilizará. Esto se puede hacer en conjunto con los usuarios del sistema. • Definir las entidades y relaciones con base en los requisitos de información identificados. • Definir las restricciones y reglas de integridad que establezcan cómo se pueden ingresar, modificar y eliminar datos de la base de datos. **Actividad 2.** • Revisar los estándares y políticas de seguridad que aplican al sistema. • Identificar las vulnerabilidades del modelo de datos actual y	• Interpreta el informe de análisis identificando las características del software a diseñar. • Define las plataformas tecnológicas a emplear en el desarrollo de acuerdo con las condiciones del software a desarrollar. • Define los entregables de diseño siguiendo los conceptos y principios de orientación a objetos. • Crea el diagrama de clases de acuerdo con los requisitos,	**Técnica:** ● Diagramas y presentar informes **Instrumento** ● Documento Respuesta: **Evidencia DR_04-01** **Evidencia DR_04-02** **Evidencia DR_04-03** **Evidencia DR_04-04** **Técnica:** ● Formulación de preguntas **Instrumento** ● Cuestionario: **Evidencia C_04-01**	***RAE: 1.*** *Elaborar los artefactos de diseño del software siguiendo las prácticas de la metodología seleccionada.*

cómo podrían ser explotadas por atacantes externos. **Actividad 3.** • Identificar las necesidades del proyecto para determinar la plataforma tecnológica más adecuada. Esto se puede hacer a través de reuniones con los stakeholders, donde se identifiquen los requisitos y las características clave del software. • Seleccionar la plataforma tecnológica adecuada más adecuada para el proyecto. Se deben considerar las necesidades del proyecto y las limitaciones presupuestarias y técnicas. • Planificar la implementación de la plataforma tecnológica. Esto puede incluir la adquisición de herramientas y recursos necesarios, así como la capacitación del equipo de desarrollo en el uso de la plataforma.	aplicando buenas prácticas de diseño orientado a objetos. • Incorpora patrones de diseño propendiendo en mejores prácticas para la codificación y mantenibilidad del software. • Define la arquitectura del software dando cumplimiento a los requisitos funcionales y no funcionales.		
EL SABER **Evidencia de Conocimiento:** **Actividad 5.** • Identificar los requisitos del entorno de	• Crea la vista de componentes para visualizar el software en fases	**Técnica:** • Diagramas y presentar informes	***RAE:*** ***2.*** *Estructurar el modelo de datos del software de acuerdo con las*

producción donde se va a desplegar el software, como la configuración del sistema operativo, las bibliotecas necesarias, el espacio en disco, entre otros. • Definir el proceso de instalación del software, incluyendo la secuencia de pasos, los archivos necesarios, las dependencias y las configuraciones requeridas. **Actividad 6.** • Identificar los artefactos a evaluar, como documentos de diseño, código fuente, pruebas y documentación del usuario. • Definir los criterios de evaluación para cada uno de los artefactos, teniendo en cuenta los requisitos y estándares establecidos para el proyecto.	avanzadas del ciclo de vida. • Crea la vista de despliegue del software para determinar condiciones de la implantación de la solución informática.	**Instrumento** • Documento Respuesta: **Evidencia DR_04-05 Evidencia DR_04-06**	*especificaciones del análisis.*
EL SABER HACER **Evidencias de Desempeño:** **Actividad 1.** • Crear el diagrama entidad-relación (ER) que muestre visualmente la estructura de la base de datos y cómo se relacionan las entidades. • Normalizar la base de datos para garantizar	Elabora listas de chequeo para verificación de la documentación de diseño • Evalúa los artefactos de diseño teniendo en	**Técnica:** • Diagramas y presentar informes **Instrumento** • Documento Respuesta: **Evidencia DR_04-07**	***RAE: 3.*** *Determinar las características técnicas de la interfaz gráfica del software adoptando estándares.* ***RAE: 4.*** *Verificar los entregables de la fase de diseño*

que la información se almacene de manera eficiente y se pueda acceder fácilmente a ella. Eliminando redundancias y estableciendo restricciones para garantizar que la información sea coherente y precisa. • Crear el modelo relaciona que especifica cómo se deben estructurar las tablas y cómo se relacionan entre sí. • Realizar pruebas y ajustes para asegurarse de que la base de datos funciona correctamente y ajustarla en caso de ser necesario. **Actividad 2.** • Aplicar técnicas de seguridad a la estructura de la base de datos con base en los estándares y políticas de seguridad y las vulnerabilidades identificadas. • Optimizar el rendimiento de la base de dato para garantizar que el sistema funcione de manera eficiente y que la información se acceda rápidamente. Se deben aplicar técnicas como la indexación adecuada y la optimización de consultas.	cuenta el cumplimiento de requisitos y la calidad de los entregables generados. • Realiza mejoras a los artefactos de diseño de acuerdo con los resultados de la evaluación. Reconoce las necesidades del contexto, según las problemáticas identificadas.	**Evidencia DR_04-08** **Evidencia DR_04-09** **Evidencia DR_04-10** **Evidencia DR_04-11** **Evidencia DR_04-12** **Evidencia DR_04-13** **Técnica:** ● Observación directa **Instrumento** ● Lista de Chequeo: **Evidencia D_04-01**	*del software de acuerdo con lo establecido en el informe de análisis.*

• Verificar la integridad de los datos para garantizar que se almacenan de manera coherente y precisa. Se deben establecer restricciones y reglas de integridad para asegurarse de que los datos ingresados sean válidos y se almacenen de manera consistente. • Realizar pruebas y ajustes en caso de ser necesario. **Actividad 3.** • Investigar las opciones de plataforma tecnológica disponibles. Esto puede incluir lenguajes de programación, frameworks, sistemas operativos, bases de datos, entre otros. • Evaluar las opciones de plataforma tecnológica, teniendo en cuenta factores como la facilidad de uso, el costo, la escalabilidad, la seguridad y la disponibilidad de recursos y herramientas de soporte. **Actividad 4.** • Identificar las clases del sistema, teniendo en cuenta las funcionalidades que se deben implementar y la relación entre ellas.			

• Agrupar las clases en paquetes asegurando que deben estar organizados de manera lógica y coherente. • Identificar las relaciones entre las clases, como la herencia, la composición o la asociación. • Representar las clases y los paquetes en un diagrama de clases y de paquetes, que permita visualizar de manera clara y organizada la estructura lógica del software. • Validar el diagrama de clases y de paquetes con los stakeholders y el equipo de desarrollo, para asegurarse de que la estructura lógica del software sea coherente y se ajuste a las necesidades del proyecto. **Actividad 5.** • Generar los paquetes de instalación del software, utilizando herramientas y técnicas que permitan crear paquetes completos y confiables, como instaladores y paquetes de actualización. • Documentar los pasos de instalación en un manual de usuario, para que los usuarios puedan instalar			

y configurar el software de manera correcta. • Realizar pruebas de instalación en un entorno de pruebas, para asegurarse que se realice de manera correcta y sin errores. **Actividad 6.** • Realizar la evaluación de los artefactos, utilizando herramientas y técnicas que permitan identificar posibles errores o incumplimientos de los requisitos y estándares establecidos. • Documentar los resultados de la evaluación en un informe, que permita identificar las áreas de mejora y las recomendaciones para corregir los errores o incumplimientos encontrados. • Implementar las correcciones de los errores o incumplimientos encontrados en la evaluación, siguiendo los procedimientos establecidos. • Realizar pruebas de validación para verificar que las correcciones implementadas cumplan			

con los requisitos y estándares establecidos.			

5. GLOSARIO DE TERMINOS

Artefacto: Es cualquier elemento o documento generado durante el proceso de desarrollo de software, como pueden ser diagramas, documentos, código fuente, entre otros.

Modelo de datos: Es una representación gráfica que muestra la estructura de una base de datos, incluyendo las tablas, las relaciones entre ellas, los campos y los tipos de datos.

Diagrama de clases: Es una representación gráfica de las clases y sus relaciones en un sistema **orientado a objetos.** Muestra la estructura de un sistema a través de la definición de las clases, sus atributos y métodos, y las relaciones entre ellas.

Diagrama de flujo de datos: Es una representación gráfica que muestra cómo se mueven los datos a través de un sistema, desde su origen hasta su destino. Permite identificar los procesos que transforman los datos, las entidades que los utilizan y las relaciones entre ellos.

Diagrama de paquetes: Es una representación gráfica que muestra la estructura de un sistema a través de la organización de las clases en paquetes o módulos. Permite visualizar las dependencias entre los paquetes y la relación entre ellos.

Diagrama de componentes: Es una representación gráfica que muestra las partes de un sistema y cómo se relacionan entre sí. Permite visualizar los componentes y las interfaces que se utilizan para comunicarse entre ellos.

Diagrama de despliegue: Es una representación gráfica que muestra cómo se distribuye el software en diferentes dispositivos o servidores. Permite visualizar la arquitectura física del sistema y la relación entre los componentes y los dispositivos donde se ejecutan.

Modelo de casos de uso: Es una representación gráfica que describe las interacciones entre los usuarios y el sistema, mostrando las acciones que realiza cada actor y la respuesta del sistema. Permite identificar los requisitos funcionales del sistema.

Modelo de arquitectura: Es una representación gráfica que muestra la estructura y organización de un sistema, incluyendo los componentes, interfaces, capas y relaciones entre ellos. Permite definir la arquitectura del sistema y su diseño.

Modelo de procesos: Es una representación gráfica que muestra el flujo de trabajo en un sistema, incluyendo las actividades, tareas y decisiones que se realizan. Permite definir los procesos y procedimientos que se siguen en el sistema.

Análisis de requerimientos: proceso de identificar y definir los objetivos y necesidades de un sistema de software.

Diagrama de clases: representación gráfica de la estructura de clases y sus relaciones en un sistema de software.

Diagrama de flujo: diagrama que representa el flujo de información y control en un proceso o algoritmo.

Diagrama de secuencia: diagrama que muestra la interacción entre los objetos de un sistema de software en el tiempo.

Diseño de la arquitectura: proceso de definir la estructura del sistema de software y sus componentes.

Diseño de la interfaz de usuario: proceso de definir la apariencia y funcionalidad de la interfaz de usuario del sistema de software.

Modelo de datos: representación de la estructura y relaciones de los datos en un sistema de software.

Patrón de diseño: solución recurrente a un problema de diseño de software.

Prototipo: versión preliminar del software que permite evaluar su funcionalidad y usabilidad.

Refactorización: proceso de mejorar el diseño y la estructura del código de software existente sin cambiar su comportamiento.

Requerimientos funcionales: especificaciones de las funciones y tareas que debe realizar el sistema de software.

Requerimientos no funcionales: especificaciones de los atributos del sistema de software, como la seguridad, la escalabilidad y la usabilidad.

UML (Unified Modeling Language): lenguaje de modelado visual utilizado para diseñar sistemas de software.

6. REFERENTES BIBLIOGRÁFICOS

Gómez, M. (2006). Introducción a la metodología de la investigación científica. Córdoba, Argentina: Brujas.

Kendall, K. &. (2011). Análisis y Diseño de Sistemas de información. México: Pearson.

Rodríguez Mogel, E. (2005). Metodología de la investigación. Villahermosa - México: Universidad de Juares, Autónoma de Tabasco.

Star, R. M. (s.f.). Principios de Sistemas de Información, enfoque Administrativo. Thomson.

WEBGRAFIA

Creative Commons SENA. (2008b, 3 marzo). NTC - ISO 9001:2008 REQUISITOS DE LA DOCUMENTACION [Ilustración]. Recuperado 19 marzo, 2020, de https://sena.territorio.la/

7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autores	Yuely Adriana Arce Arias	Instructores	Software	18 de abril de 2023

8. CONTROL DE CAMBIOS

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor					