

Contenido

Enlace video de sustentación	2
Fase 1: Crear una guía de estilo.....	3
Fase 2: Documentar API	7
Fase 3: Documentar con Markdown	11
Ejemplos de uso de Markdown:	13
Fase 4: Creación de Diagramas de flujos de procesos.....	15
Elementos de diagrama de procesos	18

Enlace video de sustentación

[Enlace de Youtube](#)

[Enlace de Drive](#) (Versión de 10 minutos (la misma de Youtube) y Versión de 20 minutos (velocidad normal))

Fase 1: Crear una guía de estilo

Universidad Nacional Abierta y a Distancia
Vicerrectoría Académica y de Investigación
Curso: Formación en la disciplina de redacción técnica de software

Guía de actividades y rúbrica de evaluación – Fase 1 Crear una guía de estilo

Tipo de actividad: grupal	
Momento de la evaluación: Inicial	
Puntaje máximo de la actividad:	
La actividad inicia el: aaa/mm/dd	La actividad finaliza el: aaa/mm/dd
<p>Con esta actividad se espera conseguir los siguientes resultados de aprendizaje:</p> <p>Con la realización de esta actividad se espera que el estudiante, aprenda que una guía de estilo de software es un conjunto de reglas y pautas para crear contenido coherente y consistente en el software. Esto incluye el uso de la terminología, el formato, la gramática, la estructura y otros aspectos que afectan la claridad y la comprensión del software entre otros conocimientos:</p> <ol style="list-style-type: none"> 1. Determinar la audiencia objetivo: Antes de crear una guía de estilo, se necesita determinar a quién va dirigida. ¿Es para escritores técnicos, desarrolladores, o ambos? Esto te ayudará a adaptar tu guía a sus necesidades específicas. 2. Definir el propósito de tu guía de estilo: ¿Por qué se está creando una guía de estilo? ¿Es para asegurar la consistencia en la escritura técnica entre diferentes equipos? ¿Es para asegurarse de que toda la escritura técnica sigue pautas específicas? Saber el propósito de la guía de estilo te ayudará a crear una guía enfocada y efectiva. 3. Determinar la estructura de la guía de estilo: La guía de estilo debe tener una estructura clara y organizada. Debe incluir secciones para cosas como gramática, puntuación, formato y tono. También se puede incluir secciones específicas para cosas como documentación de código, texto de la interfaz de usuario y mensajes de error. 	

4. Definir pautas de estilo: Una vez que tienes tu estructura en su lugar, se necesita definir las pautas de estilo. Esto incluirá cosas como el tono preferido, la estructura de la oración y las reglas de formato. Se puede utilizar guías de estilo existentes como punto de partida, como el Manual de Estilo de Microsoft o la Guía de Estilo de Documentación para Desarrolladores de Google.
5. Incorpora terminología específica de la industria: La escritura técnica de software a menudo implica el uso de terminología específica de la industria. La guía de estilo debe incluir un glosario de términos que se utilizan comúnmente en el campo. Esto ayudará a asegurar que toda la escritura técnica sea consistente y utilice la terminología correcta.
6. Incluye ejemplos: Los ejemplos son una parte importante de cualquier guía de estilo. Ayudan a ilustrar cómo deben aplicarse las pautas en situaciones de la vida real. Se pueden usar ejemplos de escritura técnica propia, así como de otras fuentes.
7. Obtener comentarios del equipo: Es importante obtener comentarios de tu equipo al crear una guía de estilo. Se puede tener ideas o conocimientos que no se han considerado. También se puede solicitar comentarios una vez que se haya creado un borrador de la guía de estilo.
8. Actualizar la guía de estilo regularmente: la guía de estilo debe ser un documento vivo que se actualice regularmente. A medida que las necesidades del equipo cambien, la guía de estilo debe reflejar esos cambios.

Para el desarrollo de esta actividad tenga en cuenta que:

- Una herramienta de procesamiento de texto (como Word)
- Acceso a internet para investigar y obtener recursos.
- Revisar la programación de encuentros sincrónicos vía webconferencia, en el tema acompañamiento docente, para planificar el tiempo para la participación en el encuentro programado.
- Revisar en los recursos disponibles de guías de estilo de software existentes y los recursos disponibles en línea.

- <https://mozilla.design/firefox/>
- <https://developer.spotify.com/documentation/design>

- <https://developers.google.com/style>
- <https://support.apple.com/guide/applestyleguide/welcome/web>

La actividad consiste en:

Actividad 1

El estudiante debe elegir y proponer en el foro un software de su preferencia (software de uso libre) para documentar.

Actividad 2

De forma colaborativa, debatir en el interior del grupo y elegir uno del software que se propuso para realizar una guía de estilo sobre el software elegido

Actividad 3

Una vez que el grupo seleccione el software para documentar, deberán crear de manera colaborativa en la creación de la guía de estilo.

deberán realizar en el foro avances a tiempo para recibir la respectiva retroalimentación por parte del tutor

La guía de estilo deberá contar con los siguientes parámetros,

- Terminología y vocabulario: Identificar la terminología y el vocabulario específicos que se utilizarán en el software y establecer reglas para su uso.
- Formato: Establecer las reglas de formato para el software, incluyendo la estructura, el diseño, el color y otros aspectos relacionados.
- Gramática: Establecer las reglas de gramática para el software, incluyendo el uso correcto de la puntuación, la ortografía y la gramática.
- Otros aspectos importantes: Identificar otros aspectos importantes a considerar, como el uso de imágenes, iconos y fuentes.

Evidencias de trabajo independiente:

Las evidencias de trabajo independiente para entregar son

1. Participación en el foro de discusión
2. Aporte de su propuesta de software para documentar

Evidencias de trabajo grupal:

1. Selección de software para la creación de guía de estilo
2. Entrega de un documento final en formato PDF o Word

Tenga en cuenta que todos los productos escritos individuales o grupales deben cumplir con las normas de ortografía y con las condiciones de presentación que se hayan definido.

En cuanto al uso de referencias considere que el producto de esta actividad debe cumplir con las normas **APA 7 Ed**

Fase 2: Documentar API

Universidad Nacional Abierta y a Distancia
Vicerrectoría Académica y de Investigación
Curso: Formación en la disciplina de redacción técnica de
software

Guía de actividades y rúbrica de evaluación – Fase 2 Documentar API

Tipo de actividad: individual Momento de la evaluación: Intermedio Puntaje máximo de la actividad:	
La actividad inicia el: aaa/mm/dd	La actividad finaliza el: aaa/mm/dd
Con esta actividad se espera conseguir los siguientes resultados de aprendizaje: <p>Con la realización de esta actividad se espera que el estudiante comprenda los siguientes temas:</p> <ol style="list-style-type: none"> 1. Entender la API: Antes de empezar a documentar una API, es importante tener una comprensión clara de cómo funciona y qué ofrece. Esto puede incluir familiarizarse con los endpoints, los parámetros y las respuestas que se esperan. 2. Identificar la audiencia: Identificar a la audiencia de la API es crucial para determinar el nivel de detalle y el enfoque que se debe tomar al documentar la API. La audiencia puede incluir a desarrolladores, usuarios finales o ambos. 3. Crear una estructura de documentación: Una estructura clara y fácil de seguir para la documentación es esencial. Esta puede incluir secciones para la introducción, la descripción general, los endpoints, los parámetros, las respuestas y los ejemplos. 4. Escribir las descripciones de los endpoints: Las descripciones de los endpoints deben incluir información detallada sobre cada uno de ellos, incluyendo el tipo de método HTTP, el formato de las solicitudes y las respuestas, y cualquier parámetro requerido. 	

5. Realizar pruebas de la API: Antes de lanzar la documentación de la API, se debe probar para asegurarse de que todo esté funcionando correctamente.

Para el desarrollo de esta actividad tenga en cuenta que:

- Necesita crear una cuenta en Postman.com (disponible en <https://www.postman.com/signup/> de forma gratuita).
- Una API para probar (puede ser una API pública o una API creada por los estudiantes para otro proyecto). También puede utilizar alguna disponible de forma gratuita:
 - <https://dev.to/ruppysupply/7-free-public-apis-you-will-love-as-a-developer-166p>
 - <https://rapidapi.com/collection/list-of-free-apis>

La actividad consiste en:

Actividad 1

Acceder al portal de Postman ([postman](https://postman.com)) y crear una cuenta allí para entender su propósito. Postman es una herramienta útil para probar y depurar APIs y que es una habilidad importante para cualquier desarrollador de software. Basados en esto vamos a documentar la API, igualmente, se va a documentar la API.

Actividad 2

Al acceder al portal, pueden usar el software en el navegador. Se debe verificar que la cuenta esté correctamente creada.

Actividad 3

Creación de solicitudes: Los estudiantes deben crear al menos dos solicitudes de API dentro de la colección. Las solicitudes pueden ser GET, POST, PUT o DELETE y deben estar relacionadas con la API seleccionada.

Actividad 4

El estudiante debe probar las solicitudes creadas para asegurarse de que la API esté funcionando correctamente. También se deben entender las respuestas de la API y verificar que coincidan con las.

Actividad 5

Documentar las solicitudes realizadas especificando cuáles métodos y parámetros se usan en la API seleccionada.

Ejemplo:

- **API:** randomuser.me
- **Estructura URL:** <https://randomuser.me/api/>
- **Método:** GET
- **Parámetros:**
 - **gender** (*string*): define si el usuario es de género masculino femenino o indefinido
 - **name** (*array*): lista de detalles del nombre
 - **title** (*string*): título del usuario, como por ejm Sr, Sra, etc.
 - **first** (*string*): Primer nombre del usuario
 - **last** (*string*): Apellido del usuario
 - **location** (*array*): ubicación del usuario
 - **street** (*array*): detalles de la dirección del usuario
 - **number** (*integer*): número de la dirección del usuario
 - **name** (*string*): nombre de la dirección del usuario, por ejm, Calle Washington.
- **Ejemplo:**

```
"results": [
  {
    "gender": "female",
    "name": {
      "title": "Mrs",
      "first": "Iida",
      "last": "Salminen"
    },
    "location": {
      "street": {
        "number": 1426,
        "name": "Hatanpään Valtatie"
```

Evidencias de trabajo independiente:

Las evidencias de trabajo independiente para entregar son

1. Participación en el foro de discusión
2. Realizar solicitudes de API y que se reciban respuesta de estas correctamente.
3. Documentar el proceso de solicitud de API especificando métodos y parámetros.

Evidencias de trabajo grupal:

1. No se requieren evidencias de trabajo grupal.

Tenga en cuenta que todos los productos escritos individuales o grupales deben cumplir con las normas de ortografía y con las condiciones de presentación que se hayan definido.

En cuanto al uso de referencias considere que el producto de esta actividad debe cumplir con las normas **APA 7 Ed**

Fase 3: Documentar con Markdown

Universidad Nacional Abierta y a Distancia
Vicerrectoría Académica y de Investigación
Curso: Formación en la disciplina de redacción técnica de
software

Guía de actividades y rúbrica de evaluación – Fase 3 Crear un documento en Markdown

Tipo de actividad: grupal Momento de la evaluación: Intermedio Puntaje máximo de la actividad:	
La actividad inicia el: aaa/mm/dd	La actividad finaliza el: aaa/mm/dd
<p>Con esta actividad se espera conseguir los siguientes resultados de aprendizaje:</p> <p>Con este ejercicio, los estudiantes de la Universidad UNAD podrán aprender los conceptos básicos de Markdown y cómo utilizarlos para dar formato al texto de manera simple. También podrán aprender a compartir su trabajo a través de GitHub, lo cual es una habilidad importante para cualquier estudiante de ingeniería de sistemas. Al realizar esta actividad se espera que el estudiante, aprenda qué es el lenguaje Markdown y usarlo de forma práctica para crear un documento que use este formato. De igual forma se busca que el estudiante use un repositorio en línea para guardar los cambios de este documento, de la misma forma que los desarrolladores guardan líneas de código dentro de un proyecto que se trabaja en conjunto. Estas son algunas habilidades que se pueden adquirir al conocer este lenguaje y el contexto de dónde aplicarlo:</p> <ul style="list-style-type: none"> • Formateo de texto: Con Markdown, el estudiante aprenderá a aplicar estilos como negrita, cursiva, títulos y enlaces a tus documentos y contenido en línea. • Estructuración de contenido: Markdown le enseñará al estudiante cómo organizar el contenido en secciones, subsecciones y listas, lo que facilita la lectura y comprensión. • Colaboración y versionado: Al utilizar Markdown en plataformas de colaboración como GitHub, el estudiante adquirirá habilidades para trabajar en equipo, realizar contribuciones y utilizar el control de versiones. 	

- Comunicación efectiva: Markdown le ayudará al estudiante a comunicar sus ideas y conceptos de manera clara y visualmente atractiva, lo que es fundamental para la comunicación efectiva en entornos técnicos y de desarrollo.

Para el desarrollo de esta actividad tenga en cuenta que:

- Necesita un editor de texto con soporte para Markdown, como stackedit.io, Typora, Visual Studio Code. Puede usar stackedit.io es gratuito.
- Debe crear una cuenta de GitHub.

La actividad consiste en:

Actividad 1

Descargar e instalar un editor de texto con soporte para Markdown, o usar un editor web.

Actividad 2

Crear un archivo Markdown nuevo en tu editor de texto.

Actividad 3

Utilizar los elementos básicos de Markdown para dar formato al texto, tales como:

- # encabezados
- *cursiva* y **negrita**
- Enlaces [link](<https://www.ejemplo.com>)
- Listas (-, 1., 2., etc)
- Imágenes ![alt text](image.jpg)

Actividad 4

Una vez que se haya creado el archivo Markdown, crear una cuenta en GitHub y crear un repositorio nuevo.

Actividad 5

Subir el archivo Markdown al repositorio nuevo en GitHub.

Evidencias de trabajo independiente:

Las evidencias de trabajo independiente para entregar son

1. Participación en el foro de discusión
2. Creación de un documento que utilice todos los elementos de formato del lenguaje Markdown
3. Crear un repositorio en Github y subir el documento allí

Evidencias de trabajo grupal:

1. Compartir y visualizar el documento de otros compañeros en Github

Tenga en cuenta que todos los productos escritos individuales o grupales deben cumplir con las normas de ortografía y con las condiciones de presentación que se hayan definido.
En cuanto al uso de referencias considere que el producto de esta actividad debe cumplir con las normas **APA 7 Ed**

Ejemplos de uso de Markdown:

Markdown es un lenguaje de marcado ligero que le permite formatear texto de una manera fácil de leer y escribir. Aquí están algunos de los conceptos básicos de la sintaxis y las características:

1. Encabezados: Use el símbolo de almohadilla (#) para crear encabezados. El número de almohadillas determina el nivel del encabezado, con una almohadilla representando el encabezado más grande.

Ejemplo:

Encabezado 1
Encabezado 2
Encabezado 3

2. Negrita e itálica: Use asteriscos (*) o guiones bajos (_) para crear texto en negrita o cursiva.

Ejemplo:

****Texto en negrita****
Texto en cursiva

3. Listas: Use guiones (-) o asteriscos (*) para crear listas no ordenadas, o use números para crear listas ordenadas.

Ejemplo:

- Elemento 1
- Elemento 2
- Elemento 3

1. Elemento 1
2. Elemento 2
3. Elemento 3

...

4. Enlaces: Use corchetes para crear un enlace, seguido de la URL entre paréntesis.

Ejemplo:

[Texto del enlace](http://www.ejemplo.com)

5. Imágenes: Use un signo de exclamación seguido de corchetes para crear una imagen, seguido de la URL de la imagen entre paréntesis.

Ejemplo:

![Texto alternativo](http://www.ejemplo.com/imagen.jpg)

6. Código: Use acentos graves (`) para formatear código dentro de una línea de texto, o use tres acentos graves para crear un bloque de código.

Ejemplo:

Esto es `código en línea`.

`Este es un bloque de código.`

7. Línea horizontal: Use tres o más guiones, asteriscos o guiones bajos para crear una línea horizontal.

Ejemplo:

Estos son solo algunos ejemplos de la sintaxis y características básicas de Markdown. Markdown también se puede usar para crear tablas, citas en bloque y más. Una vez que haya aprendido lo básico, Markdown es una excelente manera de formatear rápidamente texto para documentos, sitios web y más.

Fase 4: Creación de Diagramas de flujos de procesos
Universidad Nacional Abierta y a Distancia
Vicerrectoría Académica y de Investigación
Curso: Formación en la disciplina de redacción técnica de software

Guía de actividades y rúbrica de evaluación – Fase 4 Creación de diagramas de flujos de procesos

Tipo de actividad:	
Momento de la evaluación: Intermedio	
Puntaje máximo de la actividad:	
La actividad inicia el: aaa/mm/dd	La actividad finaliza el: aaa/mm/dd
<p>Con esta actividad se espera conseguir los siguientes resultados de aprendizaje:</p> <p>Con este ejercicio, los estudiantes de la Universidad UNAD aprenderán a crear diagramas de flujo de proceso de software efectivos y claros utilizando símbolos y notaciones estándar. También podrán mejorar su habilidad para comunicar procesos de software, lo cuales de uso común dentro de la documentación de software y la redacción técnica.</p> <p>Un flujo de proceso, también conocido como diagrama de flujo, es una herramienta visual utilizada para representar un proceso o un sistema en términos de sus componentes y cómo se relacionan entre sí. Teniendo esto en cuenta, al saber crear diagramas de proceso los estudiantes podrán:</p> <ul style="list-style-type: none"> ▪ Determinar el proceso que desean representar en su diagrama de flujo. Este podría ser un proceso de negocio, un sistema informático o cualquier otra cosa que se desee visualizar. ▪ Identificar los componentes individuales que forman parte del proceso. Estos pueden ser personas, equipos, sistemas informáticos, software o cualquier otro elemento relevante. ▪ Determinar cómo se relacionan entre sí los componentes del proceso. Por ejemplo, ¿un sistema informático interactúa con una persona en particular, o hay una secuencia específica de eventos que deben ocurrir para que el proceso se complete con éxito? ▪ Crear símbolos para representar cada componente del proceso y las relaciones entre ellos. Los símbolos comunes utilizados en los diagramas de flujo incluyen rectángulos para representar actividades o acciones, diamantes para representar decisiones, 	

círculos para representar puntos de inicio o finalización y flechas para representar flujos de trabajo.

- Crear el diagrama de flujo utilizando los símbolos que ha creado y asegúrese de que la secuencia de los símbolos sea clara y fácil de seguir. El diagrama de flujo debe tener una estructura lógica y ser fácil de entender para cualquiera que lo vea.

Para el desarrollo de esta actividad tenga en cuenta que:

- Un software de creación de diagramas como Bizagi Modeler, Lucidchart o Gliffy.

La actividad consiste en:

Actividad 1

Pensar en un proceso de software para diagramar, como la creación de un nuevo usuario en una aplicación o la revisión de código de un proyecto.

Actividad 2

Crear un diagrama de flujo para este proceso utilizando símbolos de diagrama de flujo como rectángulos para representar actividades, diamantes para representar decisiones y flechas para mostrar el flujo del proceso.

Actividad 3

Incluir detalles como entradas y salidas del proceso, así como cualquier condición especial que pueda afectar el flujo del proceso.

Actividad 4

Compartir el diagrama de flujo con tus compañeros de clase y pide retroalimentación sobre cómo mejorar la claridad y legibilidad del diagrama.

Evidencias de trabajo independiente:

Las evidencias de trabajo independiente para entregar son

1. Participación en el foro de discusión
2. Creación de un documento que utilice todos los elementos de formato del lenguaje Markdown
3. Crear un repositorio en Github y subir el documento allí








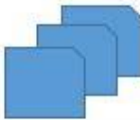

Evidencias de trabajo grupal:

1. Compartir y visualizar el documento de otros compañeros en Github

Tenga en cuenta que todos los productos escritos individuales o grupales deben cumplir con las normas de ortografía y con las condiciones de presentación que se hayan definido.

En cuanto al uso de referencias considere que el producto de esta actividad debe cumplir con las normas **APA 7 Ed**

Elementos de diagrama de procesos

SIMBOLO	SIGNIFICADO
	Operación: Se usa para describir cualquier actividad. En el interior del rectángulo se escribe una breve descripción de la actividad.
	Límites del Proceso: Indica el inicio y el final de un proceso. En el interior del eclipse aparece la palabra inicio o fin.
	Punto de Decisión: Denota que en ese punto se toma una decisión. Los outputs salidas del diamante, son siempre dos y del tipo SI / No.
	Movimiento: Muestra el movimiento de un output entre distintos puntos de la organización.
	Conector: Señala que el output de ese proceso puede ser el input de otro (la letra indica el proceso de entrada)
	Dirección del flujo: Denota la dirección y el orden de los pasos del proceso
	Documento: Documento/registro.
	Listados: Listados / notas de trabajo acumulado, información referente a la actividad.
	Base de datos: Punto de archivo donde se retiene temporalmente la información, en espera que se cumplan otras condiciones para continuar el proceso. Puede llevar asociada una tarea de administración de almacenamiento.