Practica 11. Practicar YAML

***Para practicar el lenguaje YAML se debe hacer una versión cómica y una realista sobre tu u otro personaje. Utiliza un lenguaje cuidado y adecuado a los centros públicos. Utiliza un IDE (visual estudio code) para crear y guardar tus archivos. Crea una carpeta para el proyecto YAML. Dentro ubica al menos 3 archivos de YAML. Dos de currículo y uno de intereses (véase los archivos de muestra). Crea repositorios. Sigue los pasos para subir el contenido en GitPages.***

**\*\*Práctica Guiada de YAML**

En esta práctica,

Se creará un archivo YAML para organizar información sobre sus intereses, como música, deportes y juegos. Luego, aprenderás a utilizar GitHub Pages para publicar esta información en un sitio web personalizado.

Generalmente utiliza la extensión de archivo ".yaml" o ".yml". Ambas extensiones se utilizan comúnmente para archivos YAML y son intercambiables; puedes usar cualquiera de las dos extensiones para tus archivos YAML.

Las diferencias entre ".yaml" y ".yml" son principalmente convenciones de nomenclatura y preferencias. Algunos sistemas y herramientas tienen preferencia por una extensión sobre la otra, pero en términos de contenido y estructura, los archivos YAML con ambas extensiones son idénticos.

# archivo.yaml

nombre: "Ejemplo YAML"

edad: 30

# archivo.yml

nombre: "Ejemplo YAML"

edad: 30

La practica es larga, pero, básicamente haz de coger lo que te interesa.

Es decir, tienes que obtener los siguientes pasos importantes.

Si deseas utilizar archivos HTML como estructura para tu sitio web en GitHub Pages, a continuación, tes muestran los pasos para subir estos archivos y configurar GitHub Pages para que funcione con HTML:

**### Paso 1: Crear Archivos HTML**

1. \*\*En tu Computadora:\*\*

- Crea los archivos HTML que deseas mostrar en tu sitio web. Por ejemplo, puedes tener `index.html`, `about.html`, `contact.html`, etc.

### **Paso 2: Crear un Repositorio en GitHub**

1. \*\*En GitHub:\*\*

- Crea un nuevo repositorio en tu cuenta de GitHub.

**### Paso 3: Subir Archivos HTML al Repositorio**

1. \*\*En GitHub:\*\*

- Sube los archivos HTML que creaste al repositorio en GitHub.

- Puedes hacer esto arrastrando los archivos a la interfaz de GitHub o utilizando comandos Git en tu terminal local.

**### Paso 4: Configurar GitHub Pages para Utilizar HTML**

1. \*\*En GitHub:\*\*

- Ve a la página principal de tu repositorio en GitHub.

- Haz clic en la pestaña "Settings" (Configuración) en la parte superior del repositorio.

2. \*\*Configurar GitHub Pages:\*\*

- Desplázate hacia abajo hasta la sección "GitHub Pages".

- Selecciona la rama que contiene tus archivos HTML (por lo general, `main` o `master`).

- Elige la opción "root" para publicar desde la carpeta raíz del repositorio.

- Guarda la configuración.

**### Paso 5: Verificar el Sitio Web**

1. \*\*En GitHub:\*\*

- Después de unos minutos, tu sitio web estará disponible en una URL similar a `https://nombredeusuario.github.io/nombredelrepositorio`.

- Visita esta URL en tu navegador para verificar que tu sitio web se ha desplegado correctamente.

Ahora, los archivos HTML que subiste a tu repositorio estarán disponibles en tu sitio web de GitHub Pages. Puedes enlazar entre tus diferentes páginas HTML utilizando hipervínculos estándar de HTML (`<a href="ruta/a/otra/pagina.html">Enlace</a>`). Asegúrate de que las rutas en tus enlaces sean correctas para que los usuarios puedan navegar fácilmente por tu sitio web.

#### Paso 1: Preparar el Entorno

1. \*\*Instalar Visual Studio Code:\*\* Si aún no lo tienes instalado, descarga e instala Visual Studio Code desde [https://code.visualstudio.com/](https://code.visualstudio.com/).

2. \*\*Instalar la Extensión YAML:\*\* Abre Visual Studio Code, haz clic en el icono de extensiones en la barra lateral izquierda, busca "YAML" y selecciona la extensión ofrecida por Red Hat.

#### Paso 2: Crear un Nuevo Archivo YAML

1. Abre Visual Studio Code y crea un nuevo archivo. Guárdalo con el nombre `curriculum.yaml`.

#### Paso 3: Definir la Información Personal

```yaml

informacion\_personal:

nombre: Tu Nombre

direccion: Tu Dirección

telefono: Tu Número de Teléfono

correo: Tu Correo Electrónico

```

#### Paso 4: Agregar un Resumen

```yaml

resumen: |

Soy un apasionado desarrollador de software con experiencia en varios lenguajes de programación y tecnologías web.

```

#### Paso 5: Especificar la Experiencia Laboral

```yaml

experiencia\_laboral:

- puesto: Desarrollador de Software

empresa: Empresa A

periodo: Enero 2018 - Presente

descripcion: Desarrollo y mantenimiento de aplicaciones web.

- puesto: Analista de Datos

empresa: Empresa B

periodo: Junio 2015 - Diciembre 2017

descripcion: Análisis y visualización de datos para la toma de decisiones.

```

#### Paso 6: Enumerar la Educación

```yaml

educacion:

- titulo: Licenciatura en Informática

institucion: Universidad XYZ

periodo: 2011 - 2015

- titulo: Maestría en Ciencias de la Computación

institucion: Universidad ABC

periodo: 2016 - 2018

```

#### Paso 7: Agregar Habilidades

```yaml

habilidades:

- Java

- Python

- HTML/CSS

- JavaScript

- Git/GitHub

- SQL

```

#### Paso 8: Incluir Proyectos

```yaml

proyectos:

- nombre: Proyecto A

descripcion: Desarrollo de una aplicación web para gestión de tareas.

- nombre: Proyecto B

descripcion: Implementación de un sistema de recomendación utilizando algoritmos de machine learning.

```

#### Paso 9: Añadir Idiomas y Certificaciones

```yaml

idiomas:

- idioma: Inglés

nivel: Avanzado

certificaciones:

- certificacion: Certificado de Desarrollador Web

institucion: Certifying Body

fecha: Julio 2019

```

#### Paso 10: Guardar y Revisar

Guarda el archivo `curriculum.yaml` y observa cómo la extensión YAML resalta la sintaxis para hacer que el código sea más legible. Asegúrate de que la indentación sea correcta para evitar errores.

Recuerda que este es solo un ejemplo básico. Puedes expandir y personalizar tu currículum añadiendo más detalles según sea necesario.

Esta práctica te permitirá familiarizarte con la sintaxis YAML mientras creas un currículum vitae estructurado y fácil de leer. ¡Espero que te sea útil!

### Paso 1: Crear un Archivo YAML de Intereses

1. \*\*En tu Computadora:\*\*

- Crea un archivo YAML llamado `intereses.yaml`.

- Estructura tu archivo YAML para incluir diferentes secciones como `musica`, `deportes` y `juegos`, y agrega algunos datos ficticios bajo cada sección. Por ejemplo:

```yaml

musica:

- Rock

- Hip Hop

- Electrónica

deportes:

- Fútbol

- Baloncesto

- Tenis

juegos:

- Ajedrez

- Videojuegos de rol

```

### Paso 2: Subir el Archivo YAML a GitHub

1. \*\*En GitHub:\*\*

- Crea un nuevo repositorio en tu cuenta de GitHub, por ejemplo, `mis-intereses`.

- Inicializa el repositorio con un archivo `README.md`.

- Sube el archivo `intereses.yaml` al repositorio.

### Paso 3: Crear una Página Web con GitHub Pages

1. \*\*En GitHub:\*\*

- Ve a la pestaña "Settings" de tu repositorio `mis-intereses`.

- Desplázate hacia abajo hasta la sección de GitHub Pages.

- Selecciona la rama `main` como fuente.

- Guarda la configuración. Tu sitio web estará disponible en `https://nombredeusuario.github.io/mis-intereses`.

### Paso 4: Personalizar la Página Web

1. \*\*En tu Computadora:\*\*

- Crea un archivo `index.html` en tu repositorio. Este será tu archivo principal de la página web.

- Personaliza el archivo `index.html` para mostrar tus intereses usando los datos del archivo `intereses.yaml`. Por ejemplo:

```markdown

# Mis Intereses

## Música

- {{ intereses.musica[0] }}

- {{ intereses.musica[1] }}

- {{ intereses.musica[2] }}

## Deportes

- {{ intereses.deportes[0] }}

- {{ intereses.deportes[1] }}

- {{ intereses.deportes[2] }}

## Juegos

- {{ intereses.juegos[0] }}

- {{ intereses.juegos[1] }}

```

### Paso 5: Verificar el Sitio Web

1. \*\*En GitHub:\*\*

- Espera unos minutos para que GitHub Pages construya tu sitio web.

- Visita `https://nombredeusuario.github.io/mis-intereses` para ver tu página web personalizada con tus intereses.

Esta práctica permitirá a los estudiantes practicar la estructura y la sintaxis de YAML mientras organizan información sobre sus intereses. Además, les enseñará cómo utilizar GitHub Pages para publicar contenido web de forma gratuita. ¡Espero que encuentres útil esta práctica!

**A. U**tilizar

el mismo repositorio para los archivos de interés en formato YAML y para otros archivos en formato Markdown (.md) relacionados con su currículum.

Además, pueden crear una organización y un repositorio `nombre.github.io` para mostrar estos archivos en GitHub Pages.

Aquí hay una guía paso a paso sobre cómo hacerlo:

### Paso 1: Crear una Organización y un Repositorio `nombre.github.io`

1. \*\*En GitHub:\*\*

- Crea una nueva organización desde tu cuenta.

- Dentro de la organización, crea un nuevo repositorio llamado `nombre.github.io`. Este será el repositorio para tu sitio web personal.

### Paso 2: Subir Archivos YAML y Markdown al Repositorio

1. \*\*En tu Computadora:\*\*

- Crea un archivo YAML llamado `intereses.yaml` y un archivo Markdown llamado `curriculum.md` (o cualquier otro nombre que desees) con tus datos e intereses.

2. \*\*En GitHub:\*\*

- Sube los archivos `intereses.yaml` y `curriculum.md` al repositorio `nombre.github.io`.

### Paso 3: Configurar GitHub Pages

1. \*\*En GitHub:\*\*

- Ve a la pestaña "Settings" de tu repositorio `nombre.github.io`.

- Desplázate hacia abajo hasta la sección de GitHub Pages.

- Selecciona la rama `main` y elige el archivo `index.html` (o el nombre de tu archivo Markdown) como fuente para GitHub Pages.

- Guarda la configuración.

### Paso 4: Verificar el Sitio Web

1. \*\*En GitHub:\*\*

- Espera unos minutos para que GitHub Pages construya tu sitio web.

- Visita `https://nombre.github.io` para ver tu página web personalizada con tus intereses y tu currículum.

Al utilizar el mismo repositorio, puedes organizar varios tipos de archivos en diferentes carpetas dentro del repositorio y acceder a ellos mediante rutas relativas. Por ejemplo, puedes tener una carpeta "intereses" para los archivos YAML y una carpeta "curriculum" para los archivos Markdown.

Este enfoque te permite gestionar todos tus archivos en un solo lugar y presentarlos de forma organizada en tu sitio web personal.

**B. Añadir archivos adicionales**

a tu repositorio `nombre.github.io` después de haber subido los archivos iniciales (como el currículum y los intereses), puedes seguir estos pasos:

### Paso 1: Crear y Subir el Nuevo Archivo

1. \*\*En tu Computadora:\*\*

- Crea el nuevo archivo que deseas añadir, ya sea en formato YAML, Markdown o cualquier otro formato que prefieras.

- Guarda el archivo con un nombre descriptivo, por ejemplo, `proyectos.md`.

2. \*\*En GitHub:\*\*

- Ve al repositorio `nombre.github.io` en tu cuenta de GitHub.

- Haz clic en el botón "Add file" y luego selecciona "Upload files".

- Arrastra el nuevo archivo desde tu computadora y suéltalo en el área de subida de archivos o haz clic en "Choose your files" para buscar y seleccionar el archivo desde tu computadora.

- Proporciona un mensaje de commit descriptivo en el campo "Commit changes".

### Paso 2: Confirmar el Cambio

1. \*\*En GitHub:\*\*

- Haz clic en el botón "Commit changes" para confirmar la subida del nuevo archivo al repositorio.

### Paso 3: Verificar el Cambio en el Sitio Web

1. \*\*En GitHub:\*\*

- Espera unos minutos para que GitHub Pages construya tu sitio web con el nuevo archivo.

- Visita `https://nombre.github.io/proyectos.md` (o la ruta correspondiente a tu nuevo archivo) para ver el contenido del nuevo archivo en tu sitio web.

Al seguir estos pasos, podrás añadir archivos adicionales a tu sitio web en cualquier momento. GitHub Pages automáticamente construirá el sitio web con los nuevos archivos que subas, y podrás acceder a ellos utilizando las rutas correspondientes en tu sitio web `nombre.github.io`.

OPCIONAL:

**C. Adjuntar imágenes y videos**

En YAML, no puedes añadir imágenes directamente dentro del archivo YAML como datos binarios. Sin embargo, puedes incluir URLs o rutas de archivo a las imágenes que deseas mostrar. Aquí te explico cómo hacerlo:

### Usar URLs de Imágenes en Archivos YAML:

Dentro de un archivo YAML, puedes proporcionar la URL de una imagen alojada en un servidor web. Por ejemplo:

```yaml

producto:

nombre: "Producto A"

imagen: "https://ejemplo.com/ruta/a/la/imagen.jpg"

```

En este ejemplo, la propiedad `imagen` contiene la URL de la imagen que deseas mostrar para el producto.

### Usar Rutas de Archivo Locales en Archivos YAML:

Si las imágenes están almacenadas localmente en el sistema de archivos, puedes proporcionar la ruta de archivo relativa o absoluta a la imagen dentro del archivo YAML:

```yaml

producto:

nombre: "Producto B"

imagen: "/ruta/a/la/imagen.jpg"

```

En este caso, la ruta `/ruta/a/la/imagen.jpg` es una ruta de archivo local relativa al archivo YAML.

Es importante tener en cuenta que, cuando estás utilizando YAML en un contexto específico (por ejemplo, para configuraciones en aplicaciones), las opciones para mostrar imágenes pueden variar.

Por ejemplo, en un archivo de configuración de una página web, puedes utilizar estas URLs o rutas de archivo dentro de las estructuras de datos YAML para cargar imágenes dinámicamente en tu aplicación web.

que estén almacenados en el repositorio a tu sitio web de GitHub Pages.

Para hacerlo, sigue estos pasos:

### Paso 1: Subir las Imágenes o Videos al Repositorio

1. \*\*En GitHub:\*\*

- Sube las imágenes o videos que deseas mostrar en tu sitio web al repositorio. Puedes organizarlos en carpetas si lo deseas.

### Paso 2: Referenciar las Imágenes o Videos en tus Archivos YAML o Markdown

1. \*\*En tus Archivos YAML o Markdown:\*\*

- Dentro de tus archivos YAML o Markdown, puedes referenciar las imágenes o videos utilizando rutas relativas desde la ubicación del archivo YAML o Markdown hacia los archivos de imagen o video en tu repositorio.

Por ejemplo, si tienes una imagen llamada `imagen.jpg` en una carpeta `imagenes` dentro de tu repositorio, y tu archivo YAML está en el mismo nivel que la carpeta `imagenes`, la referencia en YAML sería:

```yaml

producto:

nombre: "Producto C"

imagen: "imagenes/imagen.jpg"

```

O si tienes un video llamado `video.mp4` en una carpeta `videos`, la referencia en Markdown sería:

```markdown

![Video](videos/video.mp4)

```

### Paso 3: Desplegar el Sitio Web en GitHub Pages

1. \*\*En GitHub:\*\*

- Si ya tienes configurado tu repositorio para GitHub Pages (usando la rama `main` u otra rama como fuente), los cambios que hagas en tus archivos YAML o Markdown se reflejarán en tu sitio web una vez que se implementen.

GitHub Pages permite cargar y mostrar imágenes y videos que están en tu repositorio. Cuando el sitio web se despliega en GitHub Pages, las rutas relativas a las imágenes o videos se interpretan correctamente y se cargan en el sitio web.