

Practia 0

☰ Nota final	0
--------------	---

1. Sintaxis y uso de Markdown

Markdown es un lenguaje de marcado ligero que permite crear documentos con formato de una manera rápida y sencilla utilizando únicamente texto plano. Se utiliza ampliamente en GitHub, documentación técnica, blogs y sitios estáticos.

1.1 ¿Qué es Markdown?

Markdown fue creado por John Gruber con el objetivo de proporcionar una sintaxis fácil de leer y escribir, que pueda convertirse a HTML de manera efectiva.

Su principal ventaja es que no requiere herramientas avanzadas: cualquier editor de texto puede utilizarse.

1.2 ¿Para qué se utiliza?

Markdown se usa para:

- Documentar proyectos
- Escritura de blogs
- Crear archivos README
- Generar contenido de páginas estáticas como Hugo
- Tomar apuntes y reportes técnicos

1.3 Sintaxis básica

Encabezados

```
# Encabezado h1
## Encabezado h2
```

Encabezado h3

Negritas y cursivas

****negritas****

cursivas

******negritas y cursivas******

Listas

Lista con puntos:

- Elemento
- Elemento

Lista ordenada:

1. Primero
2. Segundo

Código

Código en línea:

```
`codigo`
```

Bloques:

```
console.log("Hola");
```

Links

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

Imágenes

![Descripción](ruta/imagen.png)

2. Uso de Git y GitHub

2.1 ¿Qué es Git?

Git es un sistema de control de versiones que permite guardar el historial completo de un proyecto, trabajar en equipo, y manejar cambios de forma ordenada y segura.

2.2 ¿Qué es GitHub?

GitHub es una plataforma que almacena repositorios Git en la nube, permitiendo:

- Respaldo del código
- Colaboración
- Publicación de proyectos
- Uso de GitHub Pages y GitHub Actions

2.3 Comandos esenciales de Git

Inicializar repositorio

```
git init
```

Configurar usuario

```
git config --global user.name "Tu Nombre"  
git config --global user.email "correo@example.com"
```

Agregar archivos

```
git add .
```

Crear un commit

```
git commit -m "Mensaje"
```

Ver estado

```
git status
```

Conectar repositorio local con GitHub

```
git remote add origin https://github.com/usuario/repositorio.git
```

Subir cambios

```
git push -u origin main
```

2.4 Cómo crear un repositorio en GitHub

1. Ir a <https://github.com>

2. Clic en **New Repository**
 3. Dar nombre al proyecto
 4. Crear el repositorio vacío
 5. Conectar con tu carpeta local usando los comandos anteriores
-

3. Generación de páginas estáticas con Hugo y GitHub Actions

3.1 ¿Qué es Hugo?

Hugo es un generador de sitios estáticos extremadamente rápido.

Permite crear blogs, portafolios, documentación y páginas sin usar bases de datos.

Ventajas:

- Muy rápido
 - Basado en Markdown
 - Fácil de mantener
 - Ideal para GitHub Pages
-

3.2 ¿Qué es GitHub Actions?

GitHub Actions es un sistema de automatización integrado en GitHub.

Permite ejecutar tareas automáticamente, como:

- Construir tu sitio Hugo
 - Compilar código
 - Publicar tu página en GitHub Pages
-

3.3 Cómo crear un sitio estático con Hugo

Instalar Hugo

Descargar desde: <https://gohugo.io>

Crear un nuevo sitio

```
hugo new site mi-sitio
```

Agregar un tema

```
git submodule add URL-del-tema themes/nombreDelTema
```

En `hugo.toml` :

```
theme = "nombreDelTema"
```

Crear una nueva página

```
hugo new posts/reporte-practica1.md
```

Ajustar contenido en Markdown

Escribes tu entrada en el archivo creado.

Construir sitio localmente

```
hugo server -D
```

3.4 Subir el sitio a GitHub

1. Inicializar Git en la carpeta del sitio
2. Conectar con un repositorio
3. Hacer push:

```
git add .  
git commit -m "Sitio Hugo"  
git push -u origin main
```

3.5 Configurar GitHub Actions para publicar en Pages

Crear archivo:

```
.github/workflows/hugo.yml
```

Contenido:

```
name: Deploy Hugo site  
  
on:  
  push:  
    branches:  
      - main  
  
jobs:  
  build-deploy:  
    runs-on: ubuntu-latest  
    steps:  
      - name: Checkout  
        uses: actions/checkout@v3  
        with:  
          submodules: true  
          fetch-depth: 0  
  
      - name: Setup Hugo  
        uses: peaceiris/actions-hugo@v2  
        with:  
          hugo-version: 'latest'
```

```
- name: Build
  run: hugo --minify

- name: Deploy
  uses: peaceiris/actions-gh-pages@v3
  with:
    github_token: ${{ secrets.GITHUB_TOKEN }}
    publish_dir: ./public
```


Después en GitHub:

Settings → Pages → "gh-pages" branch

Y tu sitio queda publicado.

4. Enlaces solicitados

 **Portafolio en GitHub:**

 *Aquí pondrás tu URL del repositorio*

 **Página estática publicada en GitHub Pages:**

 *Aquí pondrás la URL de tu sitio en Pages*

5. Conversión del archivo Markdown a PDF

Puedes convertir este archivo  a PDF mediante:

Opciones:

- Extensión *Markdown PDF* en VSCode
 - Hugo + plugin de PDF
 - Convertidores online como pandoc
-

6. Conclusión

En esta práctica se trabajó con las bases fundamentales del desarrollo web moderno a través de herramientas esenciales: Markdown para documentación, Git y GitHub para control de versiones y colaboración, y Hugo junto con GitHub Actions para la generación y despliegue automatizado de un sitio estático. Con estas herramientas se establece la estructura inicial del portafolio profesional del estudiante.