



**UNSA**

Universidad Nacional de San Agustín

**UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA**  
**FACULTAD DE DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS**  
**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS**



**“SISTEMA GESTIÓN DEL PROCESO PARA LA ACREDITACIÓN DE  
CERTIFICACIÓN INTERNACIONAL EN EDUCACIÓN”**

**CURSO:**

Programación Web 1

**DOCENTE:**

Mg. Edith Giovanna Cano Mamani

**TEMA:**

INFORME -LAB 9 (WIKIPEDIA)

**ALUMNO:**

Apaza Anahua, Roydan

**AREQUIPA – PERÚ**

**2024**

# Informe sobre la Creación de una Página de Wikipedia Básica

## I.Introducción

- **Objetivo del Informe:** El objetivo de este informe es documentar el proceso de desarrollo y las características clave de una página web básica similar a Wikipedia. Este proyecto se llevó a cabo utilizando tecnologías web estándar, incluyendo HTML, CSS y JavaScript, así como el lenguaje Markdown para la creación y edición de contenido.
- **Descripción del Proyecto:** El proyecto consiste en la creación de una página web básica con funcionalidades similares a Wikipedia utilizando tecnologías web estándar (HTML, CSS, JavaScript) y Markdown para la gestión de contenido. El propósito de esta página es permitir a los usuarios crear, editar y visualizar páginas web en formato Markdown de manera sencilla.
- 

## II.Planificación del Proyecto

- **Requisitos Iniciales:**
  - ❖ Crear una interfaz web para gestionar páginas en formato Markdown.
  - ❖ Implementar funcionalidades para crear, editar, visualizar y listar páginas.
  - ❖ Utilizar HTML, CSS y JavaScript.
- **Investigación Previa:** Se realizó una investigación sobre la estructura y funcionamiento de Wikipedia, enfocándose en la manera en que las páginas son creadas, editadas y presentadas. También se investigaron las bibliotecas disponibles para la conversión de Markdown a HTML, seleccionando `marked.js` por su eficiencia y facilidad de uso.
- **Herramientas Utilizadas:**
  - ❖ **HTML:** Para estructurar las páginas web.
  - ❖ **CSS:** Para el diseño y la apariencia visual.
  - ❖ **JavaScript:** Para añadir interactividad y gestionar la lógica del lado del cliente.
  - ❖ **Markdown:** Para la creación y edición de contenido.
  - ❖ **marked.js:** Biblioteca para convertir Markdown a HTML.

## III.Diseño de la Página

- **Estructura de la Página:**

La estructura básica de la página incluye:

- ❖ **Encabezado:** Contiene el título de la página y enlaces a las secciones principales.
  - ❖ **Cuerpo:** Se divide en varias subpáginas para crear, editar, visualizar y listar páginas.
  - ❖ **Pie de Página:** Información adicional y enlaces.
- 
- **Interfaz de Usuario:** El diseño de la interfaz de usuario se centró en la simplicidad y usabilidad. Se utilizaron colores neutros y una tipografía clara para facilitar la lectura. La disposición de los elementos sigue un flujo lógico para guiar al usuario a través de las diferentes funcionalidades.
- 
- **Funcionalidades Clave:**
    - ❖ **Crear Páginas:** Formulario para introducir el título y contenido en Markdown.
    - ❖ **Editar Páginas:** Selección y modificación de páginas existentes.
    - ❖ **Visualizar Páginas:** Mostrar el contenido convertido de Markdown a HTML.
    - ❖ **Listar Páginas:** Mostrar una lista de todas las páginas disponibles.

## IV.Desarrollo del Proyecto

- **Creación de la Página HTML:** Cada página (`index.html`, `create.html`, `edit.html`, `view.html`, `list.html`) fue creada utilizando HTML para estructurar el contenido y formularios necesarios.
- **Implementación de CSS:** El archivo `styles.css` se utilizó para definir los estilos visuales de la página, incluyendo:
  - ❖ **Diseño Responsivo:** Asegurar que la página se vea bien en diferentes dispositivos.
  - ❖ **Colores y Tipografía:** Selección de una paleta de colores y fuentes legibles.
- **Uso de JavaScript:** Se desarrollaron varios scripts para gestionar la interactividad:
  - ❖ `create.js`: Maneja la creación de páginas.
  - ❖ `edit.js`: Gestiona la edición de páginas.
  - ❖ `list.js`: Carga y muestra la lista de páginas.
  - ❖ `main.js`: Contiene funciones comunes.
  - ❖ `view.js`: Convierte el contenido Markdown a HTML para su visualización.
- **Manejo de Formularios:** Los formularios en `create.html` y `edit.html` permiten a los usuarios ingresar y modificar contenido. Los datos se almacenan en el almacenamiento local del navegador para simplificar la implementación.
- **Conversión de Markdown a HTML:** Se utilizó la biblioteca `marked.js` para convertir el contenido escrito en Markdown a HTML. Esta biblioteca se seleccionó por su facilidad de uso y rendimiento.

## V.Pruebas y Validación

- **Métodos de Prueba:** Se realizaron pruebas manuales para asegurar que todas las funcionalidades funcionen correctamente. Las pruebas incluyeron la creación, edición, visualización y listado de páginas.
- **Resultados de las Pruebas:** Las pruebas demostraron que el sistema es capaz de manejar adecuadamente las tareas básicas de gestión de páginas en Markdown. Se identificaron y solucionaron problemas menores relacionados con la visualización y el almacenamiento de datos.
- **Validación del Contenido:** Se verificó que el contenido presentado en HTML fuera una conversión fiel del contenido escrito en Markdown, asegurando la precisión y validez de las páginas.

## VI.Resultados y Conclusiones

- **Evaluación del Proyecto:** El proyecto cumplió con éxito los objetivos iniciales, proporcionando una plataforma funcional para la gestión de páginas en Markdown.
- **Beneficios y Aprendizajes:**
  - ❖ **Beneficios:** Facilidad de creación y edición de contenido en Markdown, interfaz intuitiva.
  - ❖ **Aprendizajes:** Mejor comprensión de la integración de Markdown en aplicaciones web, experiencia en el uso de `marked.js`.
- **Mejoras Futuras:**
  - ❖ **Almacenamiento en Servidor:** Implementar una base de datos para almacenar páginas.
  - ❖ **Autenticación de Usuarios:** Añadir funcionalidades de registro y autenticación.
  - ❖ **Mejoras en la interfaz:** Optimizar la UI/UX para una mejor experiencia de usuario.

## VII.Anexos

- **Código Fuente:**

HTML:

index.html

```
<!DOCTYPE html>

<html lang="es">

<head>

  <meta charset="UTF-8">

  <title>Página de Inicio</title>

  <link rel="stylesheet" href="styles.css">
```

```

</head>

<body>

    <h1>WIKIPEDIA </h1>

    <p>Nombre Completo: Roydan Apaza Anahua</p>

    <p>Curso: Programacion Web</p>

    <p>Grupo: B</p>

    <p>Descripción: Tarea de que el usuario pueda crear paginas usando formulario y lenguaje
markdown</p>

    <nav>

        <ul>

            <li><a href="list.html">Listado de Páginas</a></li>

            <li><a href="create.html">Crear Nueva Página</a></li>

        </ul>

    </nav>

</body>

```

## list.html

```

<!DOCTYPE html>

<html lang="es">

<head>

    <meta charset="UTF-8">

    <title>Listado de Páginas</title>

    <link rel="stylesheet" href="styles.css">

</head>

<body>

    <h1>Nuestras páginas de WIKI</h1>

    <ul id="page-list"></ul>

    <div id="page-details">

        <h2 id="page-title"></h2>

        <p id="page-content"></p>

    </div>

```

```
<a href="create.html">Agregar Nueva Página</a>

<script src="scripts/marked.min.js"></script>

<script src="scripts/list.js"></script>

</body>

</html>
```

## create.html

```
<!DOCTYPE html>

<html lang="es">

<head>

  <meta charset="UTF-8">

  <title>Crear Nueva Página</title>

  <link rel="stylesheet"
href="https://cdnjs.cloudflare.com/ajax/libs/github-markdown-css/5.1.0/github-markdown.min.css">

</head>

<body>

  <h1>Crear Nueva Página</h1>

  <form id="create-form">

    <label for="page-name">Nombre de la Página:</label>

    <input type="text" id="page-name" name="page-name" required>

    <br>

    <label for="page-content">Contenido de la Página:</label>

    <textarea id="page-content" name="page-content" required></textarea>

    <br>

    <button type="submit">Crear Página</button>

  </form>

  <script src="scripts/create.js"></script>

</body>

</html>
```

## edit.html

```

<!DOCTYPE html>

<html lang="es">

<head>

    <meta charset="UTF-8">

    <title>Editar Página</title>

    <link rel="stylesheet" href="styles.css">

</head>

<body>

    <h1>Editar Página</h1>

    <form id="edit-form">

        <label for="page-content">Contenido de la Página:</label>

        <textarea id="page-content" name="page-content" required></textarea>

        <br>

        <button type="submit">Guardar Cambios</button>

    </form>

    <script src="scripts/edit.js"></script>

</body>

</html>

```

## view.html

```

<!DOCTYPE html>

<html lang="es">

<head>

    <meta charset="UTF-8">

    <title>Visualizar Página</title>

    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/github-markdown-css/5.1.0/github-markdown.min.css">

</head>

<body>

    <h1 id="page-title"></h1>

    <div id="page-content"></div>

```

```

    <a href="list.html">Volver al Listado</a>

    <script src="scripts/marked.min.js"></script>

    <script src="scripts/view.js"></script>

</body>

</html>

```

## CSS:

### styles.css

```

body {

    font-family: Arial, sans-serif;

    margin: 20px;

    background-image: url('fondo.jpg');

}

h1 {

    color: #333;

}

nav ul {

    list-style-type: none;

}

nav ul li {

    display: inline;

    margin-right: 10px;

}

```

## SCRIPTS:

### create.js

```

document.getElementById('create-form').addEventListener('submit',
function(event) {

    event.preventDefault();

```



```

    const name = document.getElementById('page-name').value;

    const content =
document.getElementById('page-content').value;

    localStorage.setItem(name, content);

    window.location.href = 'view.html?name=' +
encodeURIComponent(name);

});

```

#### edit.js

```

const pageName = new
URLSearchParams(window.location.search).get('name');

const content = localStorage.getItem(pageName);

document.getElementById('page-content').value = content;

document.getElementById('edit-form').addEventListener('submit',
function(event) {

    event.preventDefault();

    const newContent =
document.getElementById('page-content').value;

    localStorage.setItem(pageName, newContent);

    window.location.href = 'list.html';

});

```

#### list.js

```

const pageList = document.getElementById('page-list');

```

```
const pageTitle = document.getElementById('page-title');

const pageContent = document.getElementById('page-content');

for (let i = 0; i < localStorage.length; i++) {

    const pageName = localStorage.key(i);

    const li = document.createElement('li');

    li.innerHTML = `

        <a href="#"
onclick="showPage('${pageName}')">${pageName}</a>

        <a href="edit.html?name=${pageName}">(E)</a>

        <a href="#" onclick="deletePage('${pageName}')">(X)</a>

    `;

    pageList.appendChild(li);
}

function showPage(name) {

    const content = localStorage.getItem(name);

    pageTitle.textContent = name;

    pageContent.textContent = content;
}

function deletePage(name) {

    localStorage.removeItem(name);

    window.location.reload();
}
```

main.js

```
// Funciones comunes si las hay
```

marked.min.js

**JS** marked.min.js

view.js

```
function convertMarkdownToHtml(markdown) {

    let html = markdown;

    // Convertir encabezados

    html = html.replace(/^# (.*)$/gm, '<h1>$1</h1>');

    html = html.replace(/^## (.*)$/gm, '<h2>$1</h2>');

    html = html.replace(/^### (.*)$/gm, '<h3>$1</h3>');

    // Convertir negritas

    html = html.replace(/\*(.*)\*/gm, '<strong>$1</strong>');

    // Convertir itálicas

    html = html.replace(/\*(.*)\*/gm, '<em>$1</em>');

    // Convertir enlaces

    html = html.replace(/\[(.*)\]\((.*)\)/gm, '<a href="$2">$1</a>');

    // Convertir listas no ordenadas

    html = html.replace(/^s*n*/gm, '<ul>\n*');

    html = html.replace(/^(\*+)*s*n\[^\*\]/gm, '$1\n</ul>\n\n$2');

    html = html.replace(/^*(.+)/gm, '<li>$1</li>');

    // Convertir listas ordenadas

    html = html.replace(/^s*n\d\./gm, '<ol>\n1.');

    html = html.replace(/^(\d\..+)*s*n\[^\d\]/gm, '$1\n</ol>\n\n$2');

    html = html.replace(/^*\d\.(+)/gm, '<li>$1</li>');

    // Convertir saltos de línea dobles en párrafos

    html = html.replace(/\n\n/g, '</p><p>');
```

```

    // Convertir saltos de línea

    html = html.replace(/\n/g, '<br>');

    // Envolver en párrafos

    html = '<p>' + html + '</p>';

    return html;
}

document.addEventListener('DOMContentLoaded', function() {

    const pageName = new URLSearchParams(window.location.search).get('name');

    const content = localStorage.getItem(pageName);

    if (pageName && content) {

        document.getElementById('page-title').textContent = pageName;

        document.getElementById('page-content').innerHTML = convertMarkdownToHtml(content);

    } else {

        document.getElementById('page-title').textContent = "Página no encontrada";

        document.getElementById('page-content').textContent = "El contenido de la página no existe.";

    }

});

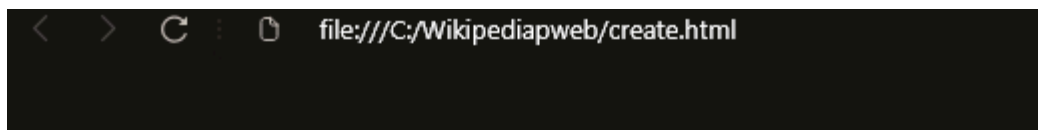
```

- **Capturas de Pantalla:**

Página de inicio



Pagina para crear páginas



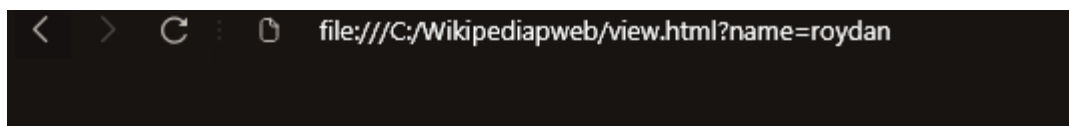
## Crear Nueva Página

Nombre de la Página:

```
**Estudiante de sistemas**  
***Cursa el segundo año de la  
carrera de Ingenieria de  
sistemas***
```

Contenido de la Página:

Pagina para mostrar la página creada



## roydan

**Estudiante de sistemas**

*Cursa el segundo año de la carrera de Ingenieria de sistemas*

[Volver al Listado](#)

Página para ver la lista de páginas



Página para editar el contenido de las páginas



- **Referencias:**

- ❖ Documentación de `marked.js`: <https://marked.js.org>
- ❖ Guía de Markdown: <https://www.markdownguide.org>