Instructions for Following Markdown (.md) Documentation Guidelines

1. Formatting Standards

- **Headers**: Use headers to structure the document hierarchically. Start with "#" for main headings and use additional "#" symbols for subheadings.

  o   For example:

    # Main Heading

    ## Subheading

    ### Sub-subheading, etc.

- **Lists**: Use "-" or "*" for bullet points, and numbers for ordered lists.

  o   **Bullets**:

    - Item 1
    - Item 2

  o   **Ordered**:

    1. Step 1

    2. Step 2

- **Code Blocks**: For code examples, use triple backticks (```) to create a code block, specifying the language if necessary:

    ```*insert programming language here*

    function example() {

        console.log("Code block example");

    }

    ```

- **Bold and Italic:** To apply these over text, you must use a specific number of "*".
  o   For example:
    ▪ For *italic*
    ▪ For **bold**
    ▪ For ***both***

- **Newlines:** To add newlines (nl), simply make a double space after a word or sentence.

- **Links:** To add links, you must use "[ ]" for link texts and followed by "()" for URL's.

Example:

For more information, make sure to follow our [website] (https://mercadocolegial.com)

2. Documenting New Features

- Each new feature added to the project must be documented with the following structure:

  o **Feature Name**: A clear and descriptive title.

  o **Description**: Explain what the feature does and why it was added.

  o **Usage Instructions**: Include code examples or command-line steps for using the feature.

  o **Related Files**: List files affected by or related to the new feature.

  Example:

  ## Feature: User Authentication

  - **Description**: Provides secure authentication using token-based methods.

  - **Usage**: Calls the `/login` endpoint to authenticate users.

  - **Related Files**: `auth.js`, `login.js`

3. Pull Requests and Code Review Process

- **Branch Naming**: Create branches with descriptive names (e.g., feature/new-login-page or bugfix/fix-login-error).

- **Commit Messages**: Write clear, concise commit messages that describe what the change does.

- Submit the pull.

For more in-depth explanations of Markdown features, check out the official GitHub Documentation on the topic or this tutorial video