

Herramientas Documentación Automática

Python — Sphinx con autodoc y napoleon

1. Instalación

```
pip install sphinx sphinx-autodoc-typehints sphinxcontrib-napoleon
```

2. Inicialización del proyecto

En la raíz del proyecto:

```
sphinx-quickstart
```

Responde a las preguntas del asistente (nombre del proyecto, autor, etc.). Esto creará una carpeta `docs/` con la configuración base.

3. Configuración del `conf.py`

Edita `docs/conf.py` y asegúrate de activar las extensiones necesarias:

```
extensions = [
    'sphinx.ext.autodoc',
    'sphinx.ext.napoleon',
    'sphinx_autodoc_typehints'
]
```

4. Generación de la documentación

```
cd docs
make html
```

La documentación generada se guardará en `docs/_build/html`.

PHP — Doctum

1. Instalación

Instala **Doctum** mediante Composer:

```
composer require --dev code-lts/doctum
```

2. Archivo de configuración (`doctum.php`)

Crea un archivo en la raíz del proyecto con este contenido:

```
<?php

use Doctum\Doctum;
use Symfony\Component\Finder\Finder;

$iterator = Finder::create()
    ->files()
    ->name('*.*')
    ->in(__DIR__ . '/src');

return new Doctum($iterator, [
    'title' => 'Documentación de SumaPHP',
    'build_dir' => __DIR__ . '/docs/build',
    'cache_dir' => __DIR__ . '/docs/cache',
]);
```

3. Generación de la documentación

```
php vendor/bin/doctum.php update doctum.php
```

La documentación se generará dentro de `docs/`.

⚡ JavaScript — JSDoc

1. Instalación

```
npm install --save-dev jsdoc
```

2. Estructura de comentarios

Asegúrate de documentar las funciones usando comentarios JSDoc:

```
/** 
 * Devuelve un objeto con el estado del semáforo.
 *
 * @param {string} color - 'red', 'yellow', 'green'
```

```

 * @returns {{red: string, yellow: string, green: string}}
 */
function obtenerEstadoSemaforo(color) {
  const estado = {
    red: "#555",
    yellow: "#555",
    green: "#555"
  };

  if (estado[color] !== undefined) {
    estado[color] = color; // Activar color correcto
  }

  return estado;
}

/**
 * Actualiza el DOM del semáforo según el estado recibido.
 *
 * @param {{red: string, yellow: string, green: string}} estado
 */
function aplicarEstadoAlDOM(estado) {
  Object.keys(estado).forEach(color => {
    const luz = document.getElementById(color);
    if (luz) luz.style.background = estado[color];
  });
}

/**
 * Función completa: calcula el estado y lo aplica al DOM
 *
 * @param {string} color
 */
function cambiarLuz(color) {
  const estado = obtenerEstadoSemaforo(color);
  aplicarEstadoAlDOM(estado);
}

// Exportación para Jest
module.exports = { obtenerEstadoSemaforo };

```

3. Generación de la documentación

Crea un archivo `jsdoc.json`:

```
{
  "source": {
    "include": ["src"]
  },
  "opts": {
    "destination": "docs/jsdoc",
  }
}
```

```
    "recurse": true
}
}
```

Ejecuta el comando:

```
npx jsdoc -c jsdoc.json
```

La documentación se generará en [docs/](#).
