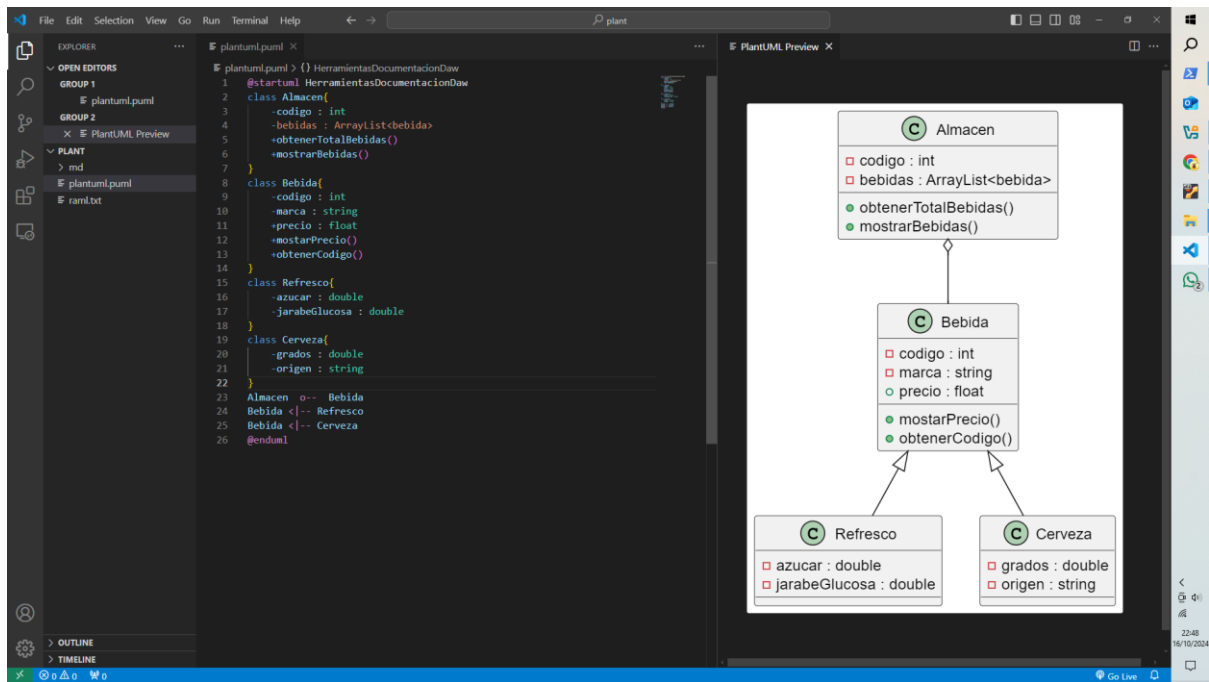


Ramón Moreno Albert 2º DAW semipresencial

## Práctica 1. Documentación con diferentes herramientas

1. Utiliza PlantUML y genera el siguiente diagrama de clases:

```
@startuml HerramientasDocumentacionDaw
class Almacen{
    -codigo : int
    -bebidas : ArrayList<bebida>
    +obtenerTotalBebidas()
    +mostrarBebidas()
}
class Bebida{
    -codigo : int
    -marca : string
    +precio : float
    +mostarPrecio()
    +obtenerCodigo()
}
class Refresco{
    -azucar : double
    -jarabeGlucosa : double
}
class Cerveza{
    -grados : double
    -origen : string
}
Almacen o-- Bebida
Bebida <|-- Refresco
Bebida <|-- Cerveza
@enduml
```



2. Utiliza MarkDown para generar el siguiente readme.md

## # Mi Proyecto Increíble

Este es un proyecto increíble que hace cosas sorprendentes.

### ## Descripción

Este proyecto realiza tareas asombrosas y útiles. Aquí hay una breve descripción de lo que puede hacer:

- \* Realiza cálculos avanzados en segundos.
- \* Genera gráficos impresionantes en tiempo real.
- \* ¡Hace tu vida más fácil!

### ## Instalación

1. Clona este repositorio: ``git clone https://github.com/tuusuario/mi-proyecto-incleible.git``
2. Ve al directorio del proyecyo: ``cd mi-proyecto-incleible``
3. Instala las dependencias: ``npm isntall``

### ## Uso

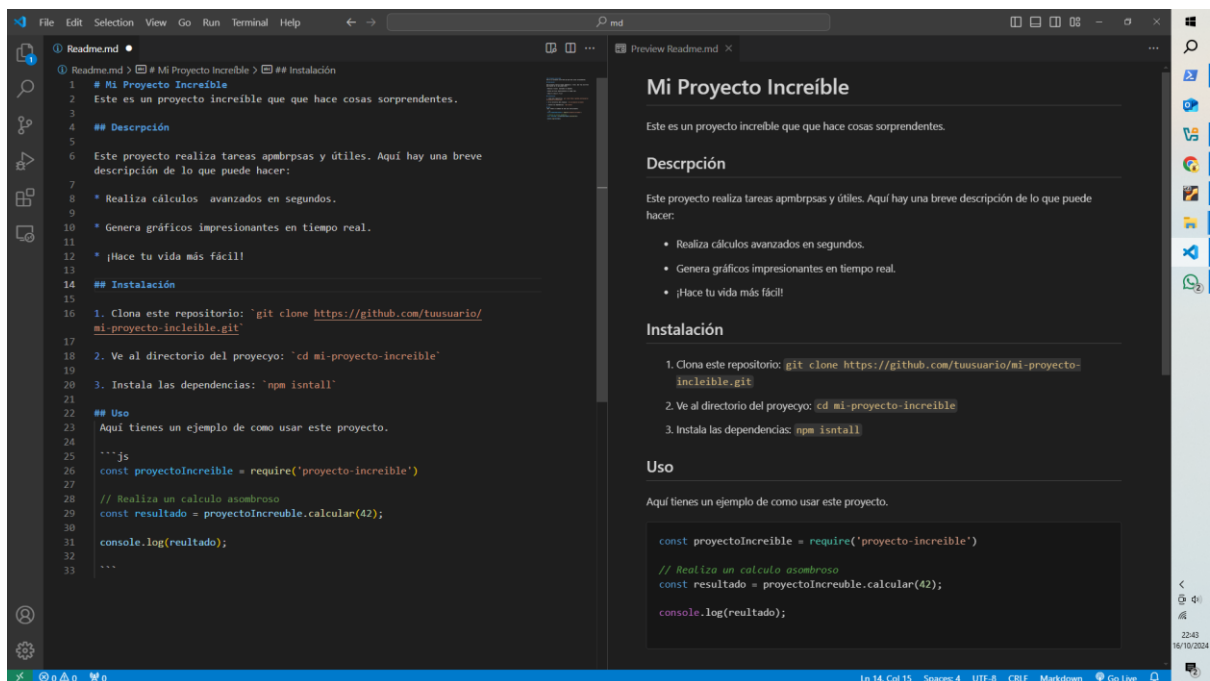
Aquí tienes un ejemplo de cómo usar este proyecto.

```
```js
const proyectoIncreible = require('proyecto-increible')

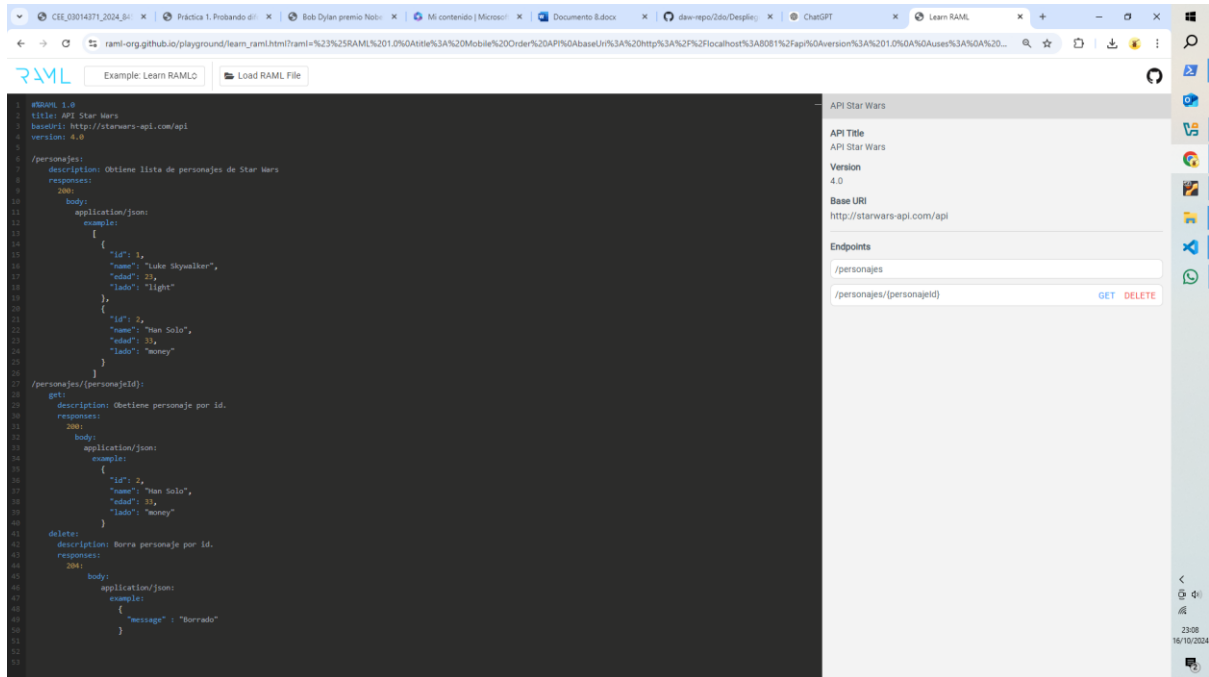
// Realiza un calculo asombroso
const resultado = proyectoIncreible.calcular(42);

console.log(reultado);

```
```



3. Utiliza RAML y adapta la siguiente documentación de la llamada API Rest que te mostro:



4. Ya que estás trabajando en PHP en la asignatura de servidor, vamos a utilizar la herramienta PHPDocumentor. El objetivo es probar la herramienta y documentar una pequeña clase. Por lo tanto, investiga cómo instalar y utilizar esa herramienta, programa una pequeña clase (puedes obtenerla de internet) y muestra la documentación que te ha generado la herramienta.

Voy a generar la documentación de una clase que he creado para la realización de este ejercicio, mediante PHPDocumentor.

PHPDocumentor genera documentación a partir de los comentarios PHPDoc en tu código. Analiza las clases, métodos y propiedades documentadas, creando archivos HTML que describen la estructura, tipos de datos y funcionalidades de tu aplicación, facilitando su comprensión.

Lo voy a usar usando Docker con los siguientes comandos:

Primero instalo Docker:

```
sudo apt update
```

```
sudo apt install docker.io
```

descargar la imagen de phpDocumentor con:

```
sudo docker pull phpdoc/phpdoc
```

Y para finalizar, situado en la carpeta donde está el proyecto que deseo documentar ejecuto el siguiente código:

```
sudo docker run --rm -v $(pwd):/data phpdoc/phpdoc
```

Esto genera la documentación en un archivo llamado index.html, en la siguiente secuencia de imágenes se puede observar la clase creada y la documentación obtenida:

La clase:

```
<?php

/**
 * Clase que representa un curso de Despliegue de Aplicaciones Web.
 *
 */
class DespliegueAplicacionesWeb
{
    /**
     * @var string Nombre de la asignatura.
     */
    private $nombre;

    /**
     * @var string Descripción de la asignatura.
     */
    private $descripcion;

    /**
     * @var array Lista de alumnos inscritos en la asignatura.
     */
    private $alumnos;

    /**
     * Constructor de la clase.
     *
     * @param string $nombre Nombre de la asignatura.
     * @param string $descripcion Descripción de la asignatura.
     */
    public function __construct($nombre, $descripcion)
    {
        $this->nombre = $nombre;
        $this->descripcion = $descripcion;
        $this->alumnos = [];
    }

    /**
     * Obtiene el nombre de la asignatura.
     *
     * @return string Nombre de la asignatura.
     */
    public function getNombre()
    {
        return $this->nombre;
    }
}
```

```

/**
 * Obtiene la descripción de la asignatura.
 *
 * @return string Descripción de la asignatura.
 */
public function getDescripcion()
{
    return $this->descripcion;
}

/**
 * Agrega un alumno a la lista de inscritos.
 *
 * @param string $alumno Nombre del alumno.
 */
public function agregarAlumno($alumno)
{
    $this->alumnos[] = $alumno;
}

/**
 * Obtiene la lista de alumnos inscritos.
 *
 * @return array Lista de alumnos.
 */
public function getAlumnos()
{
    return $this->alumnos;
}

/**
 * Muestra la información completa de la asignatura.
 *
 * @return string Información de la asignatura.
 */
public function mostrarInfo()
{
    $info = "Asignatura: " . $this->nombre .
        "\nDescripción: " . $this->descripcion .
        "\nAlumnos: " . implode(' ', $this->alumnos);
    return $info;
}

```

```
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL

• daw-semi@daw-semi-VirtualBox:~/Escritorio/phpDocc/src$ sudo docker pull phpdoc/phpdoc
[sudo] contraseña para daw-semi:
Using default tag: latest
latest: Pulling from phpdoc/phpdoc
Digest: sha256:57ff58163a6c3c3a737902aa577e6cc7a247f1af16e066d12c15c12a5055b07d
Status: Image is up to date for phpdoc/phpdoc:latest
docker.io/phpdoc/phpdoc:latest

• daw-semi@daw-semi-VirtualBox:~/Escritorio/phpDocc/src$ sudo docker run --rm -v $(pwd):/data phpdoc/phpdoc
phpDocumentor 3.4.2-v3.5.3+10b8a76

Parsing files

1/1 [=====] 100%
Applying transformations (can take a while)

All done in 0 seconds!
• daw-semi@daw-semi-VirtualBox:~/Escritorio/phpDocc/src$
```

← → ↺

file:///home/daw-semi/Escritorio/phpDocc/src/phpdoc/build/classes/DespliegueAplicacionesWeb.html

🔍 ☆

Packages

Application

Reports

Deprecated

Errors

Markers

Indices

Files

Application

# DespliegueAplicacionesWeb

in package Application

Clase que representa un curso de Despliegue de Aplicaciones Web.

Table of Contents

Properties

🔴 \$alumnos : array<string|int, mixed>

🔴 \$descripcion : string

🔴 \$nombre : string

Methods

🔴 \_\_construct() : mixed

Constructor de la clase.

🔴 agregarAlumno() : mixed

Agrega un alumno a la lista de inscritos.

🔴 getAlumnos() : array<string|int, mixed>

Obtiene la lista de alumnos inscritos.

🔴 getDescripcion() : string

Obtiene la descripción de la asignatura.

🔴 getNombre() : string

Obtiene el nombre de la asignatura.

🔴 mostrarInfo() : string

Muestra la información de la asignatura.

On this page

Table Of Contents

[Properties](#)

[Methods](#)

Properties

[\\$alumnos](#)

[\\$descripcion](#)

[\\$nombre](#)

Methods

[\\_\\_construct\(\)](#)

[agregarAlumno\(\)](#)

[getAlumnos\(\)](#)

[getDescripcion\(\)](#)

[getNombre\(\)](#)

[mostrarInfo\(\)](#)

← → ↺

file:///home/daw-semi/Escritorio/phpDocc/src/phpdoc/build/classes/DespliegueAplicacionesWeb.html

🔍 ☆

## Properties

\$alumnos

DespliegueDeAplicacionesWeb.php : 22

private array<string|int, mixed> \$alumnos

Lista de alumnos inscritos en la asignatura.

\$descripcion

DespliegueDeAplicacionesWeb.php : 17

private string \$descripcion

Descripción de la asignatura.

\$nombre

DespliegueDeAplicacionesWeb.php : 12

private string \$nombre

Nombre de la asignatura.

## Methods

\_\_construct()

DespliegueDeAplicacionesWeb.php : 30

Constructor de la clase.

public \_\_construct(string \$nombre, string \$descripcion) : mixed

On this page

Table Of Contents

[Properties](#)

[Methods](#)

Properties

[\\$alumnos](#)

[\\$descripcion](#)

[\\$nombre](#)

Methods

[\\_\\_construct\(\)](#)

[agregarAlumno\(\)](#)

[getAlumnos\(\)](#)

[getDescripcion\(\)](#)

[getNombre\(\)](#)

[mostrarInfo\(\)](#)