



**CICLO: DAW 2º AÑO  
MÓDULO DE DWES**

# **[Tarea N° 1C - RA09]**

**Alumno:  
Alejandro Jesus Vega  
Dominguez  
[75914715H]**

*Los documentos, elementos gráficos, vídeos, transparencias y otros recursos didácticos incluidos en este contenido pueden contener imprecisiones técnicas o errores tipográficos. Periódicamente se realizan cambios en el contenido. Fomento Ocupacional FOC SL puede realizar en cualquier momento, sin previo aviso, mejoras y/o cambios en el contenido.*

*Es responsabilidad del usuario el cumplimiento de todas las leyes de derechos de autor aplicables. Ningún elemento de este contenido (documentos, elementos gráficos, vídeos, transparencias y otros recursos didácticos asociados), ni parte de este contenido puede ser reproducida, almacenada o introducida en un sistema de recuperación, ni transmitida de ninguna forma ni por ningún medio (ya sea electrónico, mecánico, por fotocopia, grabación o de otra manera), ni con ningún propósito, sin la previa autorización por escrito de Fomento Ocupacional FOC SL.*

*Este contenido está protegido por la ley de propiedad intelectual e industrial. Pertenece a Fomento Ocupacional FOC SL los derechos de autor y los demás derechos de propiedad intelectual e industrial sobre este contenido.*

*Sin perjuicio de los casos en que la ley aplicable prohíbe la exclusión de la responsabilidad por daños, Fomento Ocupacional FOC SL no se responsabiliza en ningún caso de daños indirectos, sean cuales fueren su naturaleza u origen, que se deriven o de otro modo estén relacionados con el uso de este contenido.*

© 2022 Fomento Ocupacional FOC SL todos los derechos reservados.

## Contenido

<b>1. Documentos que se adjuntan a este informe.</b>	<b>1</b>
<b>2. Resto de epígrafes que componen el desarrollo de este informe.</b>	<b>1</b>
ENUNCIADO DE LA TAREA	2
RA8_d) Se han creado repositorios específicos a partir de información existente en Internet y en almacenes de información.	2
RA8_f) Se han programado servicios y aplicaciones web utilizando como base información y código generados por terceros.	2
RA8_h) Se han probado, depurado y documentado las aplicaciones generadas.	4

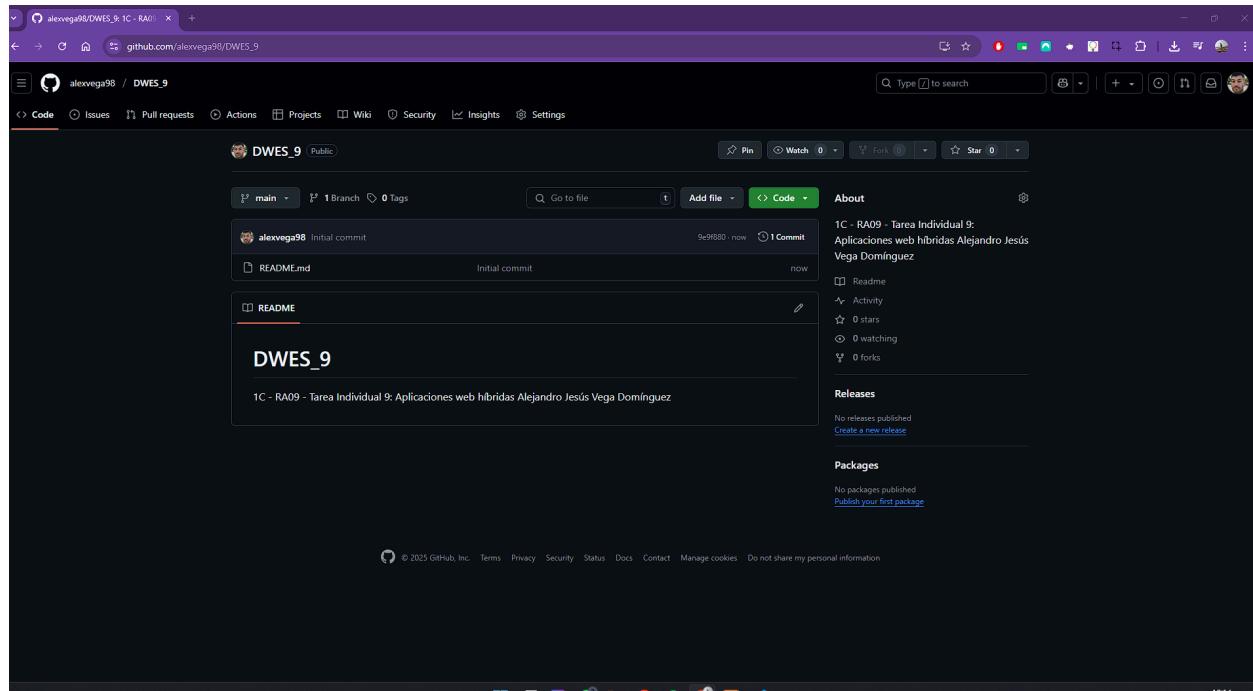
### ***ENUNCIADO DE LA TAREA***

Esta tarea consiste en realizar las siguientes operaciones:

**RA8\_d) Se han creado repositorios específicos a partir de información existente en Internet y en almacenes de información.**

- **Publicar el código fuente desarrollado en la tarea en un repositorio, por ejemplo github o Bitbucket. Incluir la url del repositorio.**

Creamos un repositorio en Github [https://github.com/alexvega98/DWES\\_9.git](https://github.com/alexvega98/DWES_9.git)



## RA8\_f) Se han programado servicios y aplicaciones web utilizando como base información y código generados por terceros.

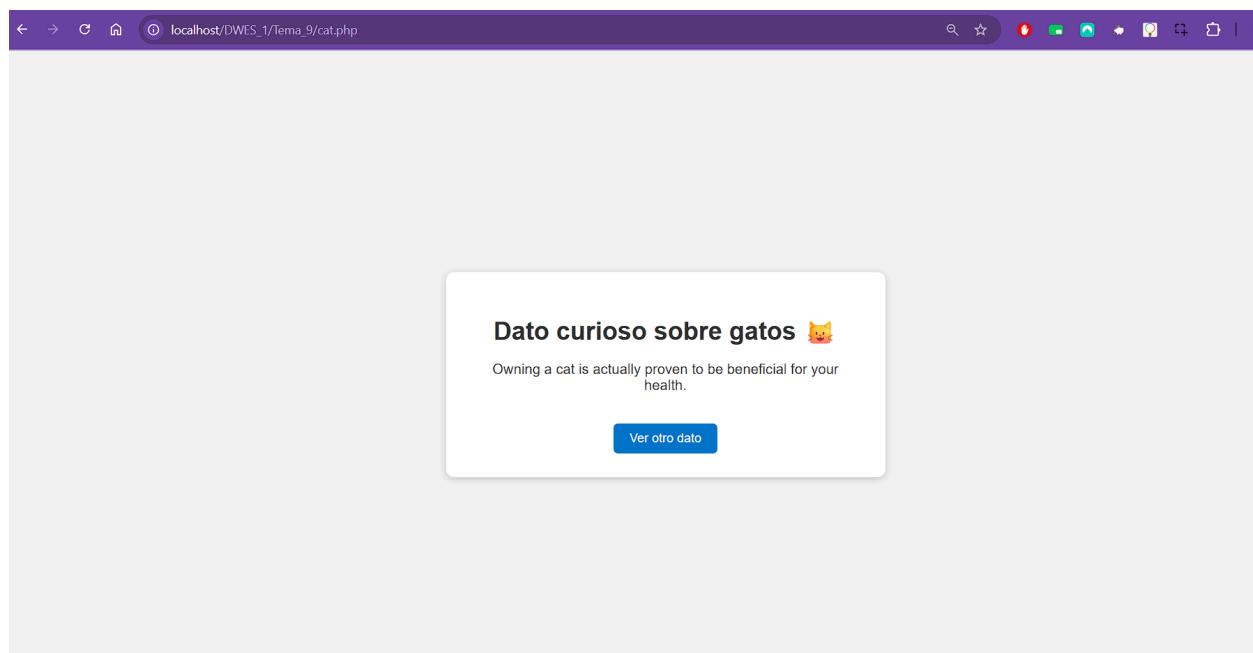
- Utilizar un servicio web y mostrar el resultado en una página web. Se puede utilizar el comando `file_get_contents()` o `curl`. Para buscar servicios de uso libre, por ejemplo, se puede utilizar la siguiente página:
  - <https://desarrolloweb.com/colecciones/api-rest-uso-publico-libre>
  - Se valorará el aspecto gráfico de la aplicación.
  - Nota: Se recomienda probar un servicio que devuelva datos en json.

En mi caso voy a usar una api sobre Cat\_fact ya que me gustan los gatos y tengo dos y me ha llamado la atención para ver como es:

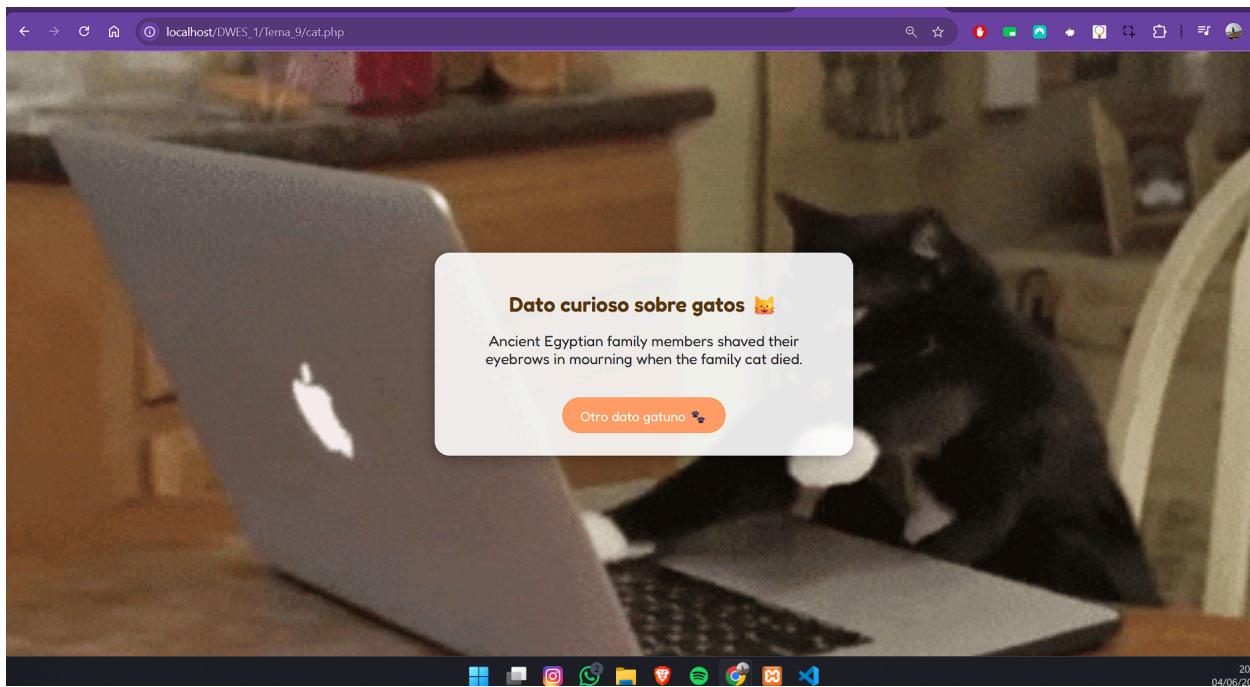
<https://github.com/alexwohlbruck/cat-facts>

Creamos el código que llama a la API y nos lo dé devuelve con un JSON

```
Bienvenido cat.php x
cat.php
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4      <meta charset="UTF-8">
5      <title>Dato Curioso de Gatos</title>
6  </head>
7  <body>
8
9  <div class="card">
10     <h1>Dato curioso sobre gatos 🐱</h1>
11
12     <?php
13     $json = file_get_contents("https://catfact.ninja/fact");
14     $data = json_decode($json, true);
15
16     if (isset($data["fact"])) {
17         echo "<p>" . htmlspecialchars($data["fact"]) . "</p>";
18     } else {
19         echo "<p>No se pudo obtener el dato.</p>";
20     }
21     ?>
22
23     <a href="cat.php">Ver otro dato</a>
24
25
26 </div>
27
28 </body>
29 </html>
30
```



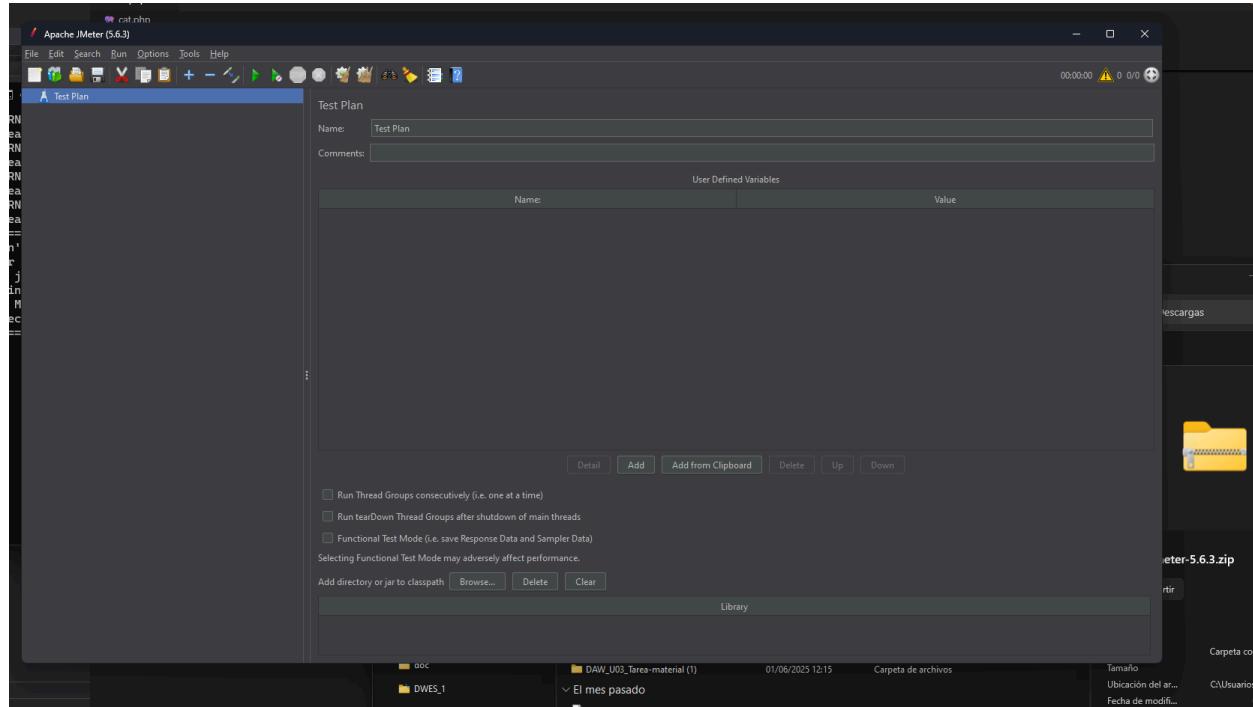
El api nos funciona bien, ahora vamos a darle aspecto de gato



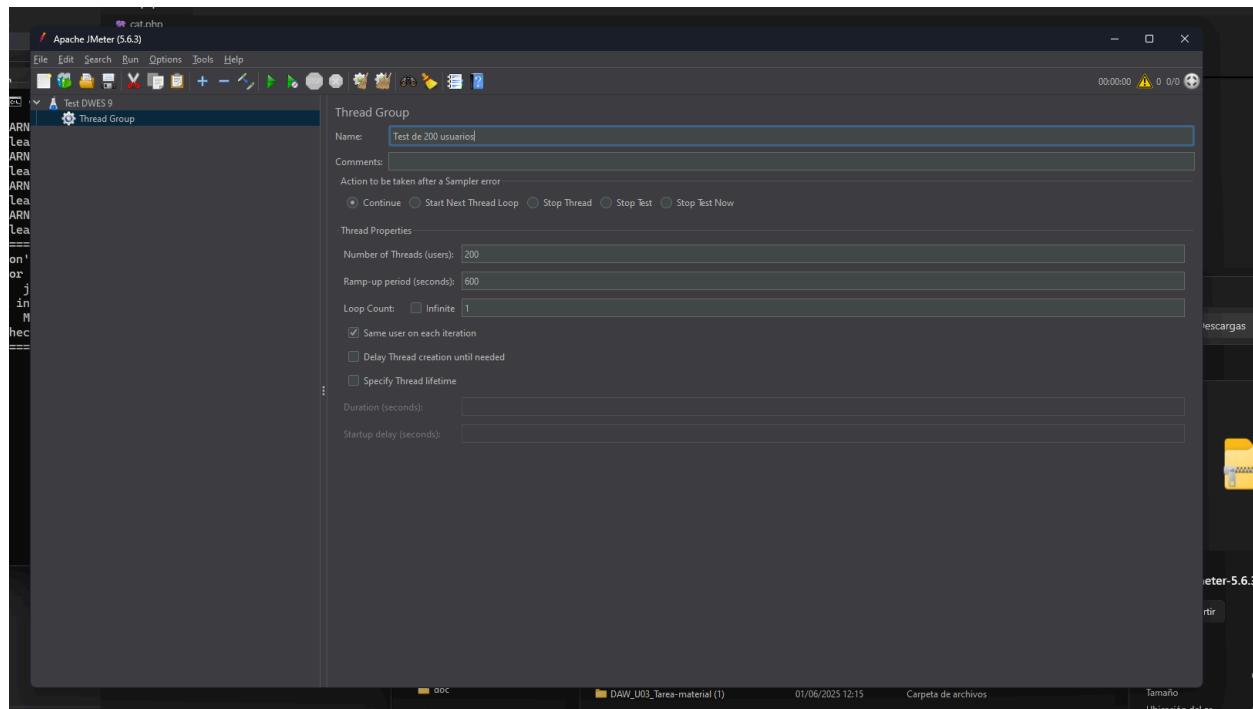
## RA8\_h) Se han probado, depurado y documentado las aplicaciones generadas.

- Desarrollar una página sencilla que permita probar la aplicación, y navegar por todas las páginas de todos los apartados de la tarea. Se valorará el aspecto gráfico de la solución.
- Llevar a cabo una prueba con JMeter, utilizando el componente Concurrency Thread Group, con la siguiente configuración:
  - Target Concurrency: número de usuarios totales que realizaran peticiones. 200 usuarios.
  - Ramp Up Time: tiempo durante el cual se irán añadiendo usuarios hasta alcanzar el total. 10 min.
  - Ramp-Up Steps Count: número de incrementos de usuarios en el tiempo definido en el punto anterior. 10 usuarios.
  - Hold Target Rate Time: tiempo durante el cual, una vez alcanzado el número total de usuarios, éstos van a permanecer realizando peticiones. 5 min.

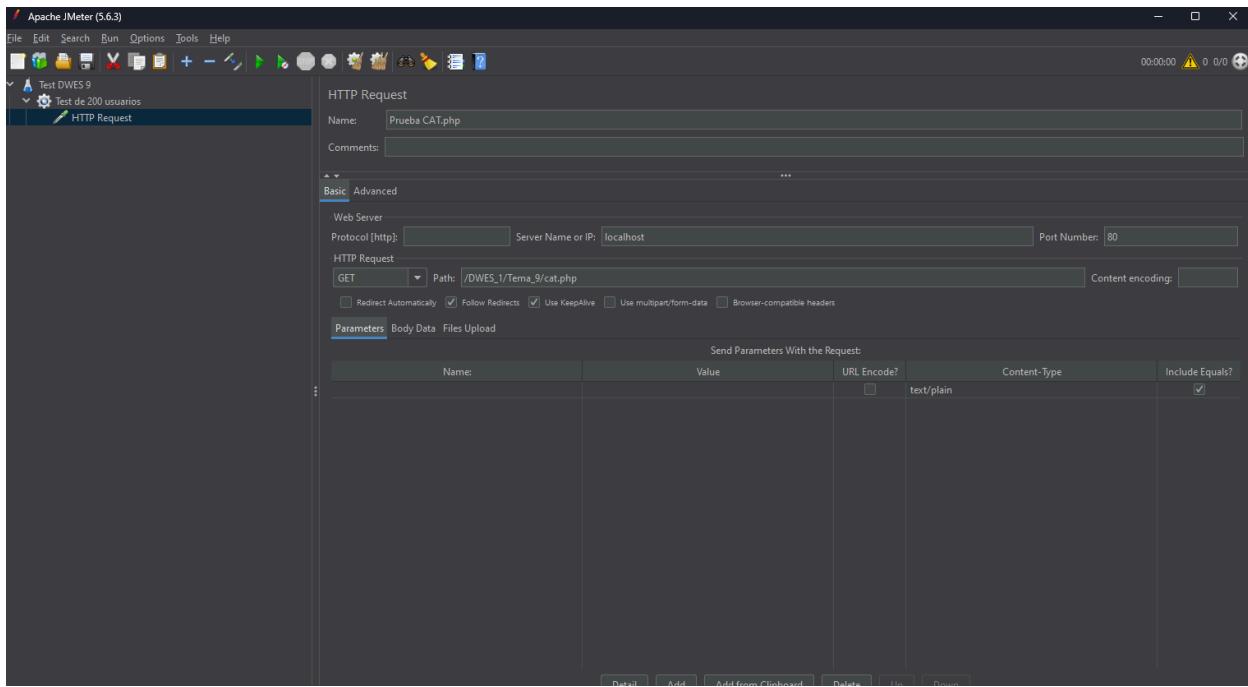
Ejecutamos el archivo jmeter.bat y nos aparece el programa



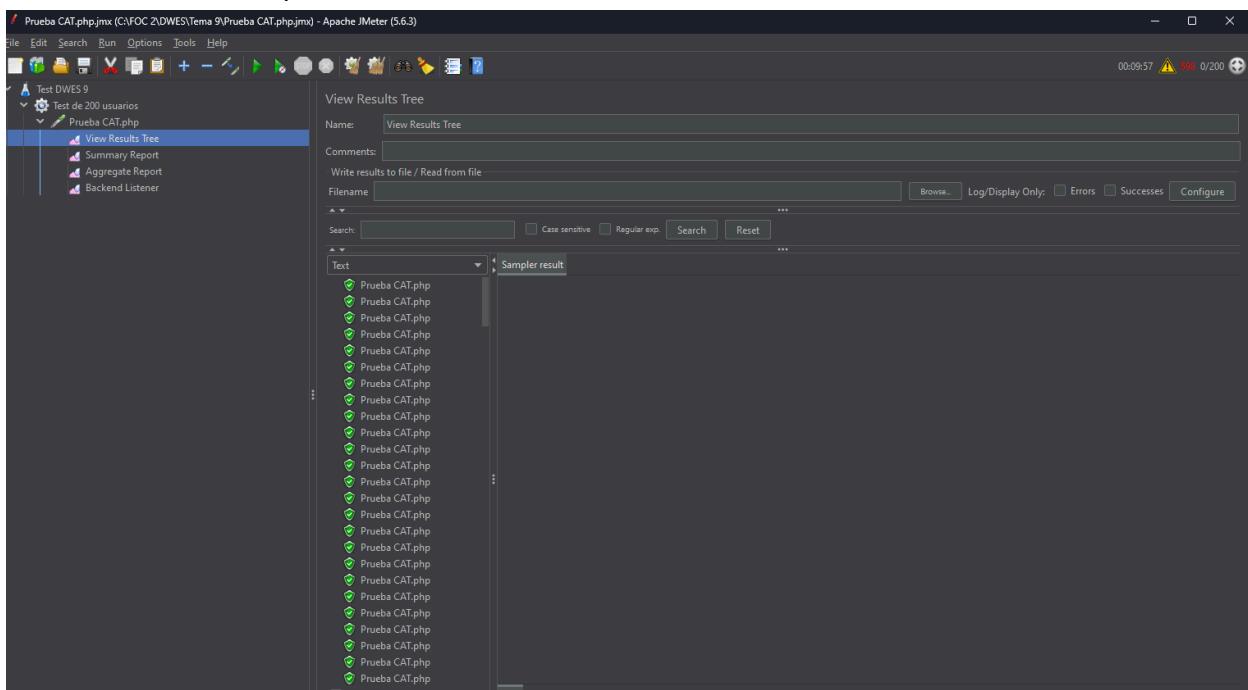
Ahora vamos a añadir los parametros que nos piden el ejercicio



Luego añadimos nuestro archivo cat.php



una vez terminada la prueba



podemos verificar que cat.php soporta muchas visualizaciones sin fallo.

Luego nos dirigimos a la carpeta con el archivo de la memoria

```

<?xml version="1.0" encoding="UTF-8"?>
<jmeterTestPlan version="1.2" properties="5.0" jmeter="5.6.3">
<hashTree>
<TestPlan guiclass="TestPlanGui" testclass="TestPlan" testname="Test DWES 9">
<elementProp name="TestPlan.user_defined_variables" elementType="Arguments" guiclass="ArgumentsPanel" testclass="Arguments" testname="User Defined Variables">
<collectionProp name="Arguments.arguments"/>
</elementProp>
</TestPlan>
<hashTree>
<ThreadGroup guiclass="ThreadGroupGui" testclass="ThreadGroup" testname="Test de 200 usuarios">
<intProp name="ThreadGroup.num_threads">200</intProp>
<intProp name="ThreadGroup.ramp_time">600</intProp>
<boolProp name="ThreadGroup.same_user_on_next_iteration">true</boolProp>
<stringProp name="ThreadGroup.on_sample_error">continue</stringProp>
<elementProp name="ThreadGroup.main_controller" elementType="LoopController" guiclass="LoopControlPanel" testclass="LoopController" testname="Loop Controller">
<stringProp name="LoopController.loops">1</stringProp>
<boolProp name="LoopController.continue_forever">false</boolProp>
</elementProp>
</ThreadGroup>
<hashTree>
<HTTPSamplerProxy guiclass="HttpTestSampleGui" testclass="HTTPSamplerProxy" testname="Prueba CAT.php">
<stringProp name="HTTPSampler.domain">localhost</stringProp>
<stringProp name="HTTPSampler.port">80</stringProp>
<stringProp name="HTTPSampler.path">/DWES_1/Tema_9/cat.php</stringProp>
<boolProp name="HTTPSampler.follow_redirects">true</boolProp>
<stringProp name="HTTPSampler.method">GET</stringProp>
<boolProp name="HTTPSampler.use_keepalive">true</boolProp>
<boolProp name="HTTPSampler.postBodyRaw">false</boolProp>
<elementProp name="HTTPSampler.Arguments" elementType="Arguments" guiclass="HTTPArgumentsPanel" testclass="Arguments" testname="User Defined Variables">
<collectionProp name="Arguments.arguments">
<elementProp name="" elementType="HTTPArgument">
<boolProp name="HTTPArgument.always_encode">false</boolProp>
<stringProp name="Argument.value"></stringProp>
<stringProp name="Argument.metadata"></stringProp>
<boolProp name="HTTPArgument.use_equals">true</boolProp>
</elementProp>
</collectionProp>
</elementProp>
</HTTPSamplerProxy>
<hashTree>
<ResultCollector guiclass="ViewResultsFullVisualizer" testclass="ResultCollector" testname="View Results Tree">
<boolProp name="ResultCollector.error_logging">false</boolProp>
<objProp>
<name>saveConfig</name>
<value class="SampleSaveConfiguration">
<time>true</time>
<latency>true</latency>

```

Ln 175, Col 72 | 10.614 caracteres. | 100% | Unix (LF) | 24/5/2025, 18:51:37 - Audio - voz receta | 24/05/2025, 18:51 | Archivo WEBM | 717 KB.

- Utilizando la sintaxis básica de PHPDoc, comentar las funciones creadas. Crear un comentario `/** */` en cada función que incluya un comentario sobre la función
- Utilizar `@param` para describir los parámetros que recibe la función y `@return` para comentar el valor resuelto
- Se puede verificar el correcto funcionamiento de los comentarios creados usando PHPDocumentor.
- Descargar el fichero `phpdocumentor.phar` de la página de PHPDocumentor
- Crear un directorio `doc` fuera del directorio donde se esté creando la página web
- Ejecutar `c:\xampp\php\bin\php.exe phpDocumentor.phar -d c:\directorio de trabajo -t c:\directorio de documentación creado.` Estos directorios son de ejemplo..
- Tras esto se debería haber creado una página web con el contenido de la documentación.

Mi código original no tenía función pero lo he añadido sin problema quedando así:

The screenshot shows a code editor interface with a dark theme. On the left, there's an 'EXPLORADOR' sidebar with sections for 'EDITORES ABIERTOS' and 'TEMA\_9'. The main area displays a file named 'cat.php' with the following content:

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <title>Dato Curioso de Gatos</title>
    <style>...
</head>
<body>
<php
/*
 * Realiza una petición a la API catfact.ninja y devuelve un dato curioso sobre gatos.
 *
 * @return string|null Devuelve el texto del dato si la API responde correctamente, o null si falla.
 */
function obtenerDatosGato() {
    $json = file_get_contents("https://catfact.ninja/fact");
    $data = json_decode($json, true);

    return $data["fact"] ?? null;
}

$dato = obtenerDatosGato();
?>

<main class="card">
    <main class="cat-ears"></main>
    <h1>Dato curioso sobre gatos <img alt="cat icon" style="vertical-align: middle;"></h1>

    <?php if ($dato): ?>
        <p><?= htmlspecialchars($dato) ?></p>
    <?php else: ?>
        <p>No se pudo obtener el dato.</p>
    <?php endif; ?>

        <a href="cat.php">Otro dato gatuno <img alt="arrow icon" style="vertical-align: middle;"></a>
</main>
</body>
</html>
```

descargamos el archivo a traves del cmd

[http://alexvegadomingue.byethost14.com/DWES\\_T9/doc/index.html](http://alexvegadomingue.byethost14.com/DWES_T9/doc/index.html)

The screenshot shows a web browser window with the URL 'alexvegadomingue.byethost14.com/DWES\_T9/doc/index.html'. The page title is 'Documentation'. On the left, there's a sidebar with navigation links: 'Documentation', 'Packages', 'Application', 'Reports', 'Deprecated', 'Errors', 'Markers', 'Indices', and 'Files'. The main content area has a heading 'Documentation' and a 'Table of Contents' section. Under 'Functions', there is a link to 'obtenerDatosGato(): string|null'. A tooltip for this function reads: 'Realiza una petición a la API catfact.ninja y devuelve un dato curioso sobre gatos.' Below this, there's a 'Return values' section with the following text: 'string|null — Devuelve el texto del dato si la API responde correctamente, o null si falla.'