



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. Pertenecen 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

1. Documentos que se adjuntan a este informe.	1
2. Resto de epígrafes que componen el desarrollo de este informe.	1
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

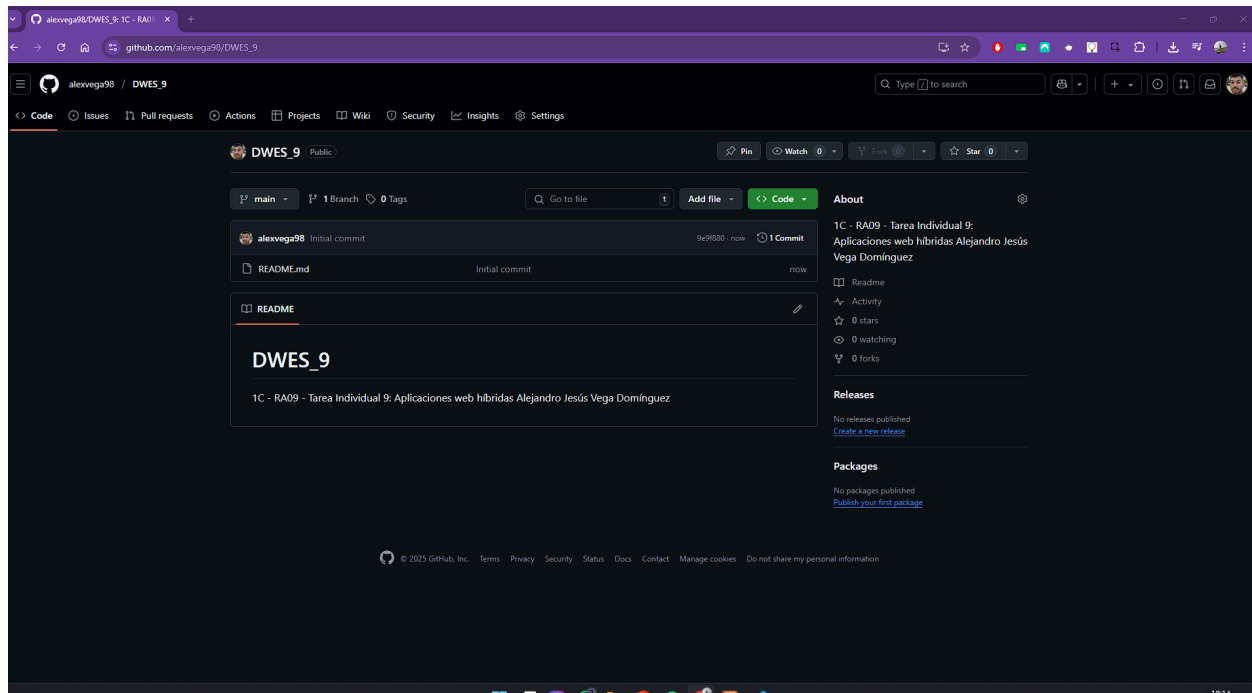
ENUNCIADO DE LA TAREA

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



RA8_f) Se han programado servicios y aplicaciones web utilizando como base información y código generados por terceros.

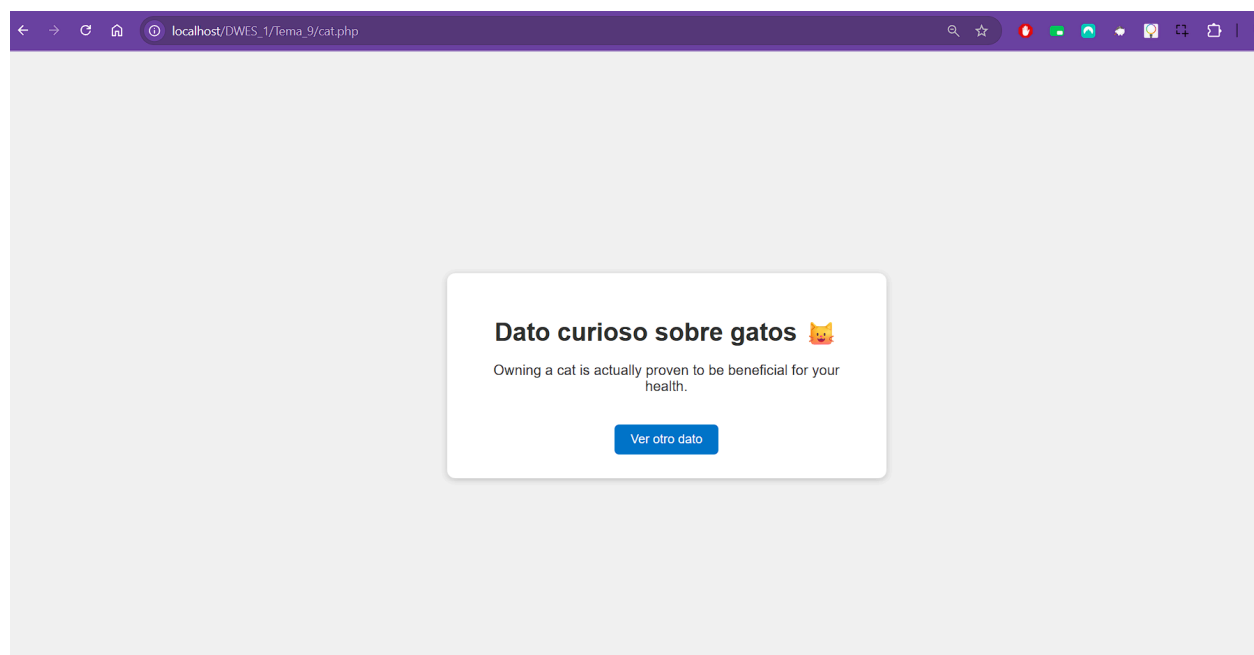
- Utilizar un servicio web y mostrar el resultado en una página web. Se puede utilizar el comando `file_get_contents()` o `curl`. Para buscar servicios de uso libre, por ejemplo, se puede utilizar la siguiente página:
 - <https://desarrolloweb.com/colecciones/api-rest-uso-publico-libre>
 - Se valorará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

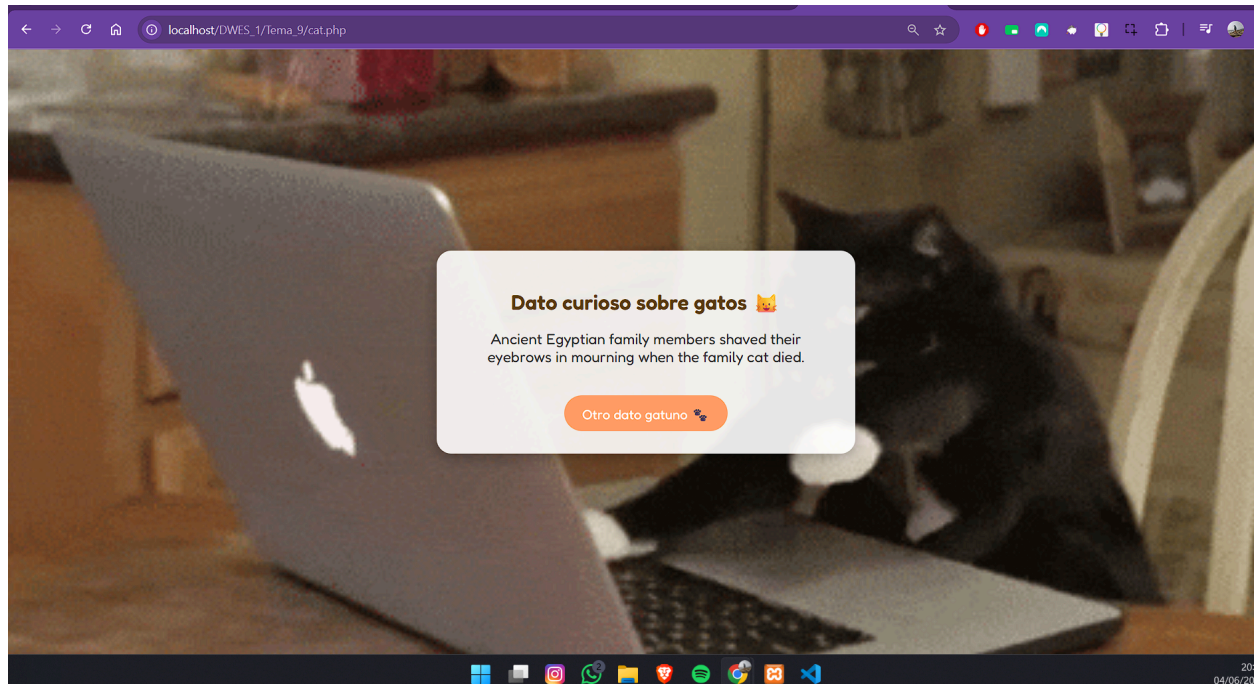
<https://github.com/alexwohlbruck/cat-facts>

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

```
cat.php
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4   <meta charset="UTF-8">
5   <title>Dato Curioso de Gatos</title>
6   <style>...
48 </style>
49 </head>
50 <body>
51
52 <div class="card">
53   <h1>Dato curioso sobre gatos 🐱</h1>
54
55   <?php
56     $json = file_get_contents("https://catfact.ninja/fact");
57     $data = json_decode($json, true);
58
59     if (isset($data["fact"])) {
60       echo "<p>" . htmlspecialchars($data["fact"]) . "</p>";
61     } else {
62       echo "<p>No se pudo obtener el dato.</p>";
63     }
64   <?>
65
66   <a href="cat.php">Ver otro dato</a>
67 </div>
68
69 </body>
70 </html>
71
```



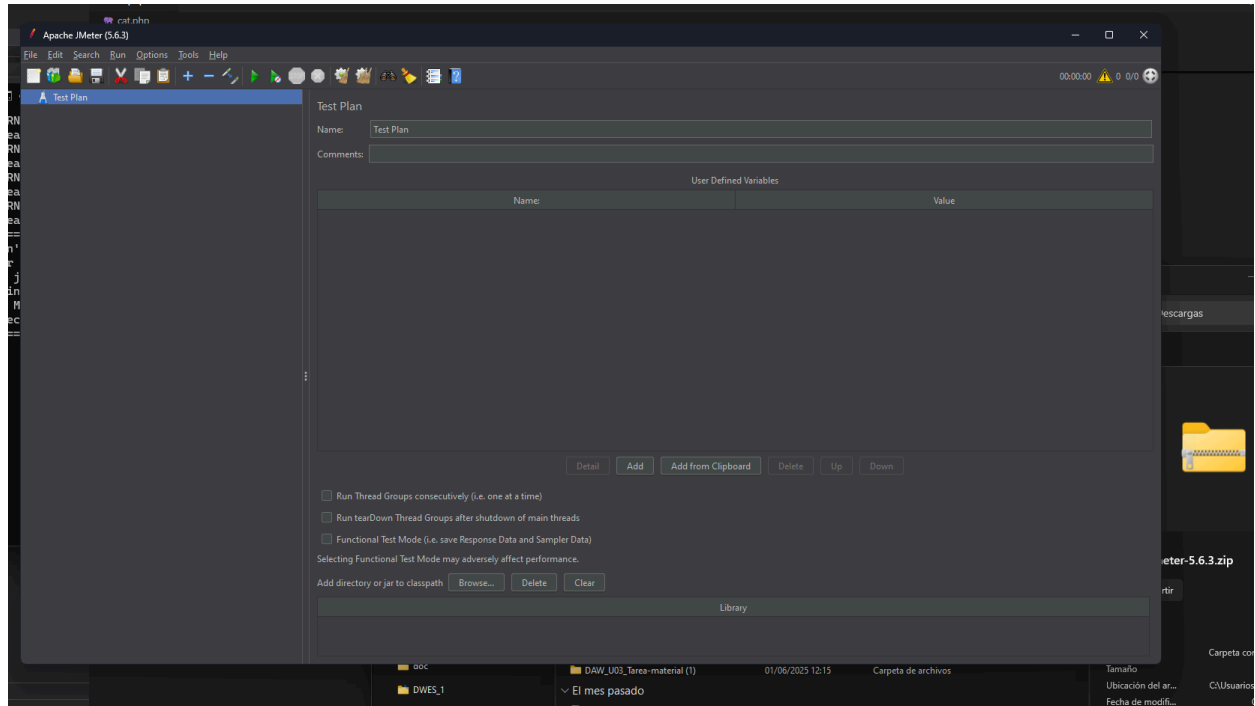
El api nos funciona bien, ahora vamos a darle aspecto de gato



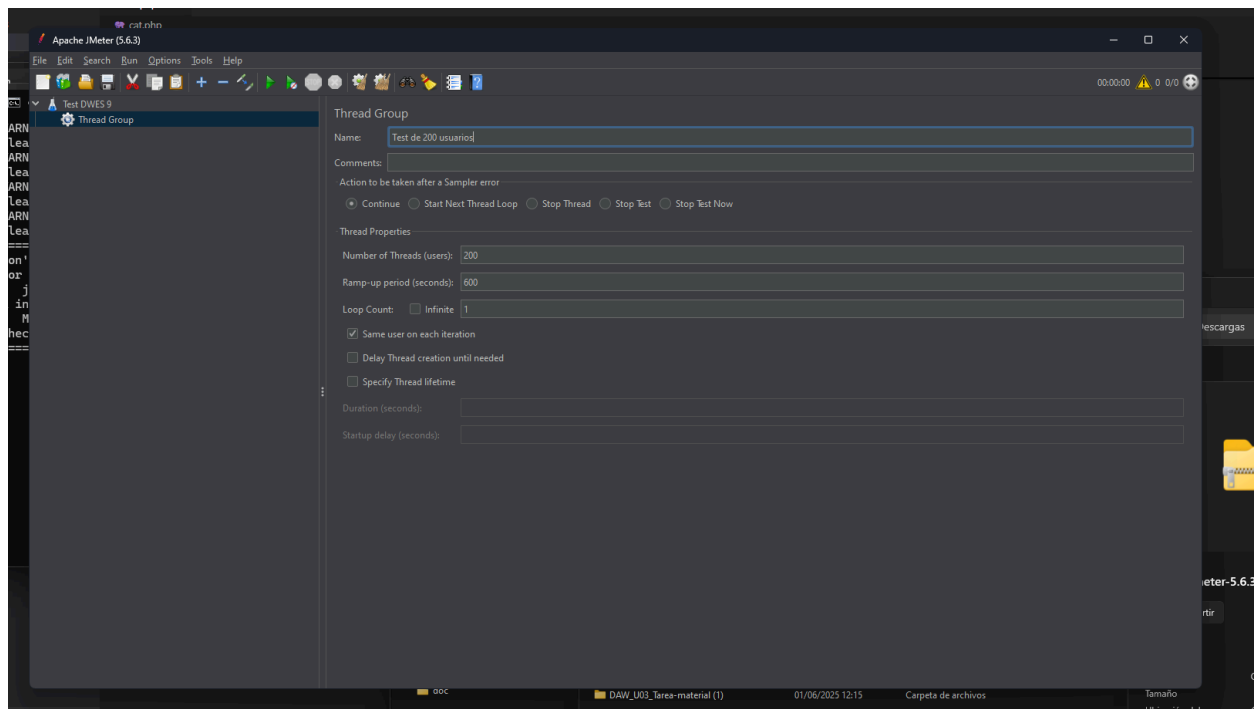
RA8_h) Se han probado, depurado y documentado las aplicaciones generadas.

- 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



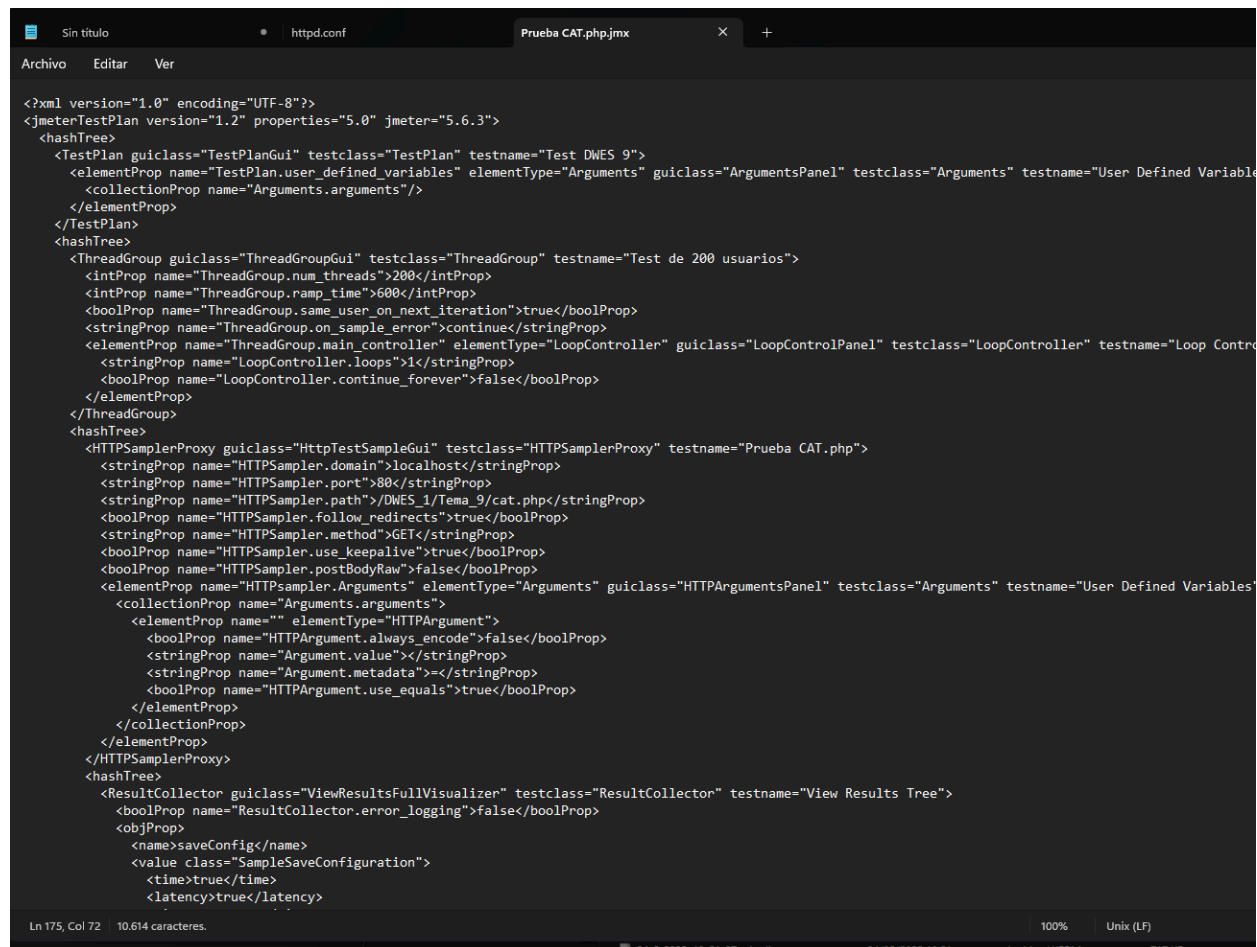
Ahora vamos a añadir los parametros que nos piden el ejercicio



Luego añadimos nuestro archivo cat.php



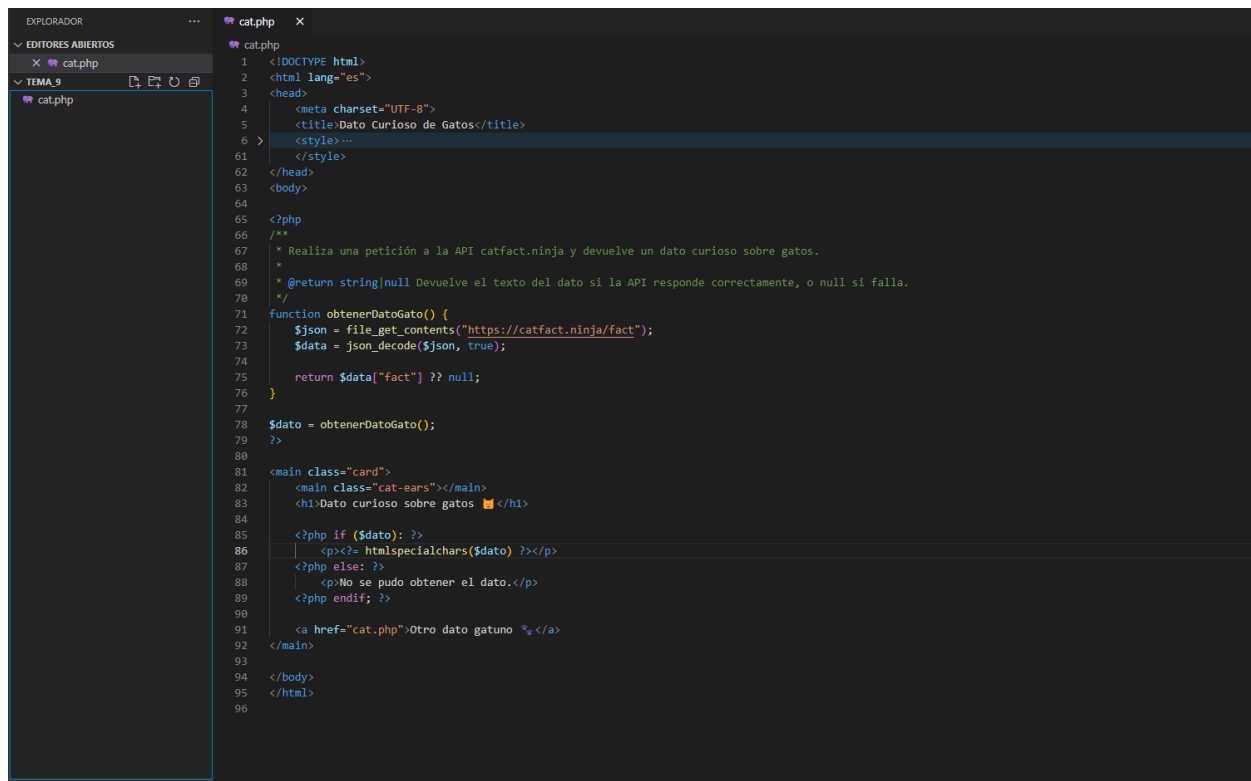
Luego nos dirigimos a la carpeta con el archivo de la memoria

A screenshot of a text editor window showing a JMeter test plan XML file named 'Prueba CAT.php.jmx'. The XML structure includes a 'TestPlan' with a 'Test de 200 usuarios' thread group, an 'HTTPSamplerProxy' for 'Prueba CAT.php', and a 'ResultCollector' for 'View Results Tree'. The 'TestPlan' has properties for version, encoding, and jmeter version. The 'ThreadGroup' has properties for number of threads (200), ramp time (600), and a loop controller with 1 loop. The 'HTTPSamplerProxy' has properties for domain (localhost), port (80), path (/DWES_1/Tema_9/cat.php), and method (GET). The 'ResultCollector' has properties for error logging (false) and latency (true).

```
<?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>
              </value>
            </objProp>
          </ResultCollector>
        </hashTree>
      </hashTree>
    </TestPlan>
  </hashTree>
</jmeterTestPlan>
```

- 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.

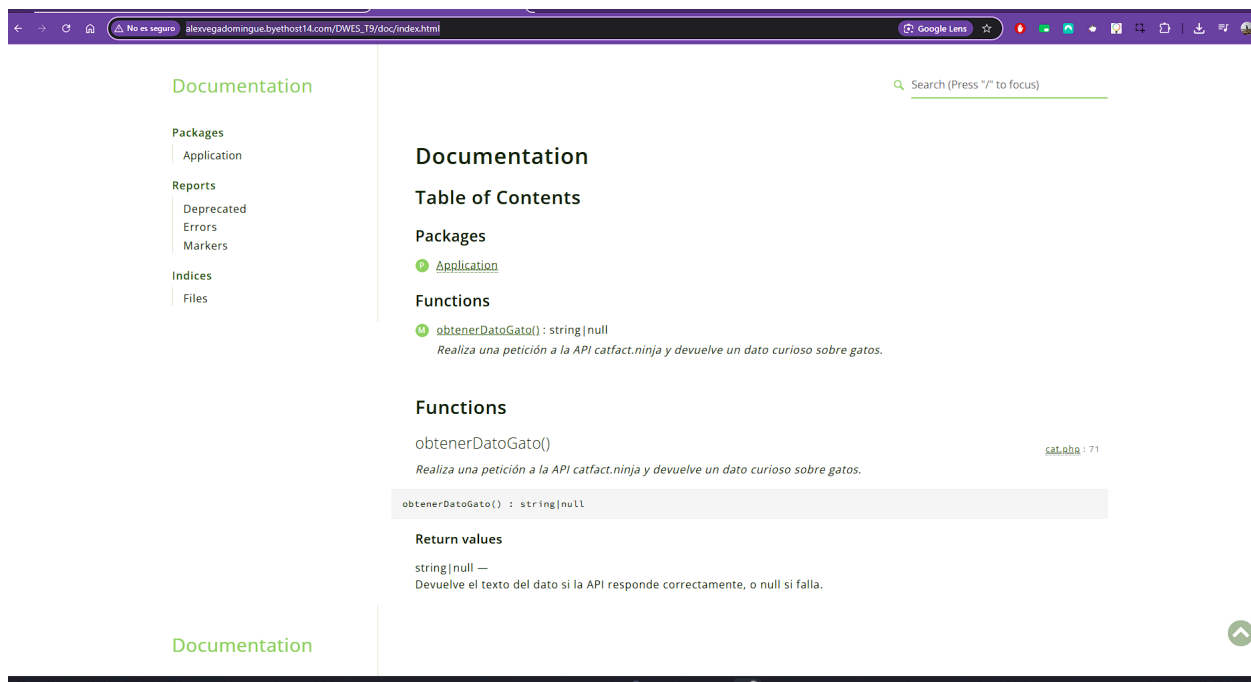
Mi código original no tenía función pero lo he añadido sin problema quedando así:



```
1 <!DOCTYPE html>
2 <html lang="es">
3 <head>
4     <meta charset="UTF-8">
5     <title>Dato Curioso de Gatos</title>
6     <style>...
61 </style>
62 </head>
63 <body>
64
65 <?php
66 /**
67  * Realiza una petición a la API catfact.ninja y devuelve un dato curioso sobre gatos.
68  *
69  * @return string|null Devuelve el texto del dato si la API responde correctamente, o null si falla.
70  */
71 function obtenerDatoGato() {
72     $json = file_get_contents("https://catfact.ninja/fact");
73     $data = json_decode($json, true);
74
75     return $data["fact"] ?? null;
76 }
77
78 $dato = obtenerDatoGato();
79 ?>
80
81 <main class="card">
82     <main class="cat-ears"></main>
83     <h1>Dato curioso sobre gatos 🐾</h1>
84
85     <?php if ($dato): ?>
86         <p><?= htmlspecialchars($dato) ?></p>
87     <?php else: ?>
88         <p>No se pudo obtener el dato.</p>
89     <?php endif; ?>
90
91     <a href="cat.php">Otro dato gatuno 🐾</a>
92 </main>
93
94 </body>
95 </html>
96
```

descargamos el archivo a traves del cmd

http://alexvegadomingue.byethost14.com/DWES_T9/doc/index.html



Documentation

Packages

- Application

Reports

- Deprecated
- Errors
- Markers

Indices

- Files

Documentation

Table of Contents

Packages

- Application

Functions

- obtenerDatoGato(): string|null
Realiza una petición a la API catfact.ninja y devuelve un dato curioso sobre gatos.

Functions

obtenerDatoGato()

Realiza una petición a la API catfact.ninja y devuelve un dato curioso sobre gatos.

obtenerDatoGato(): string|null

Return values

string|null —
Devuelve el texto del dato si la API responde correctamente, o null si falla.