Metodología para contribuir a *Apache Software Foundation*

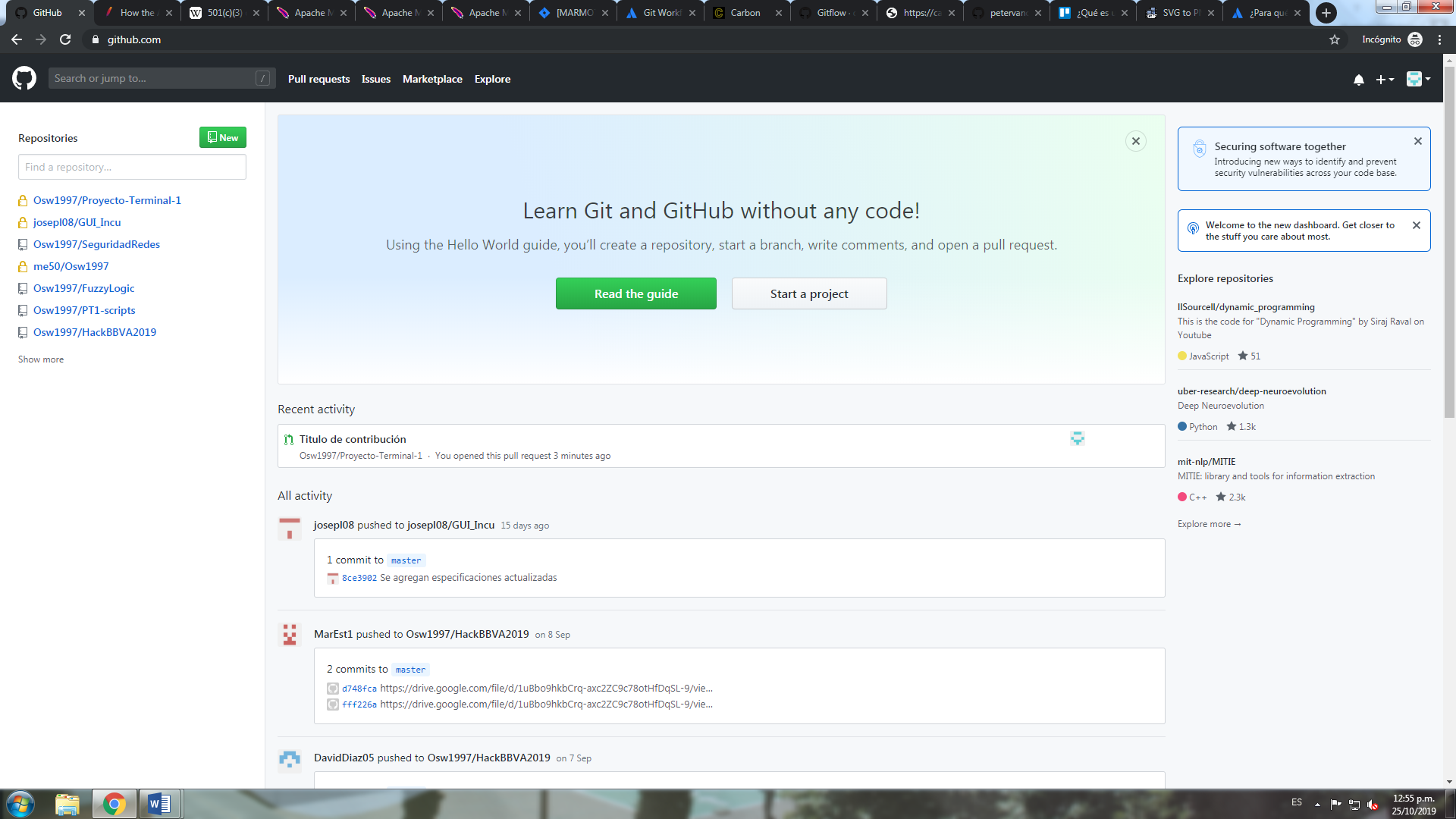
Objetivo: Definir los pasos a seguir para contribuir a la organización *Apache Software Foundation.*

Resultados esperados: Documento que indique el protocolo a seguir para contribuir a *Apache Software Foundation.*

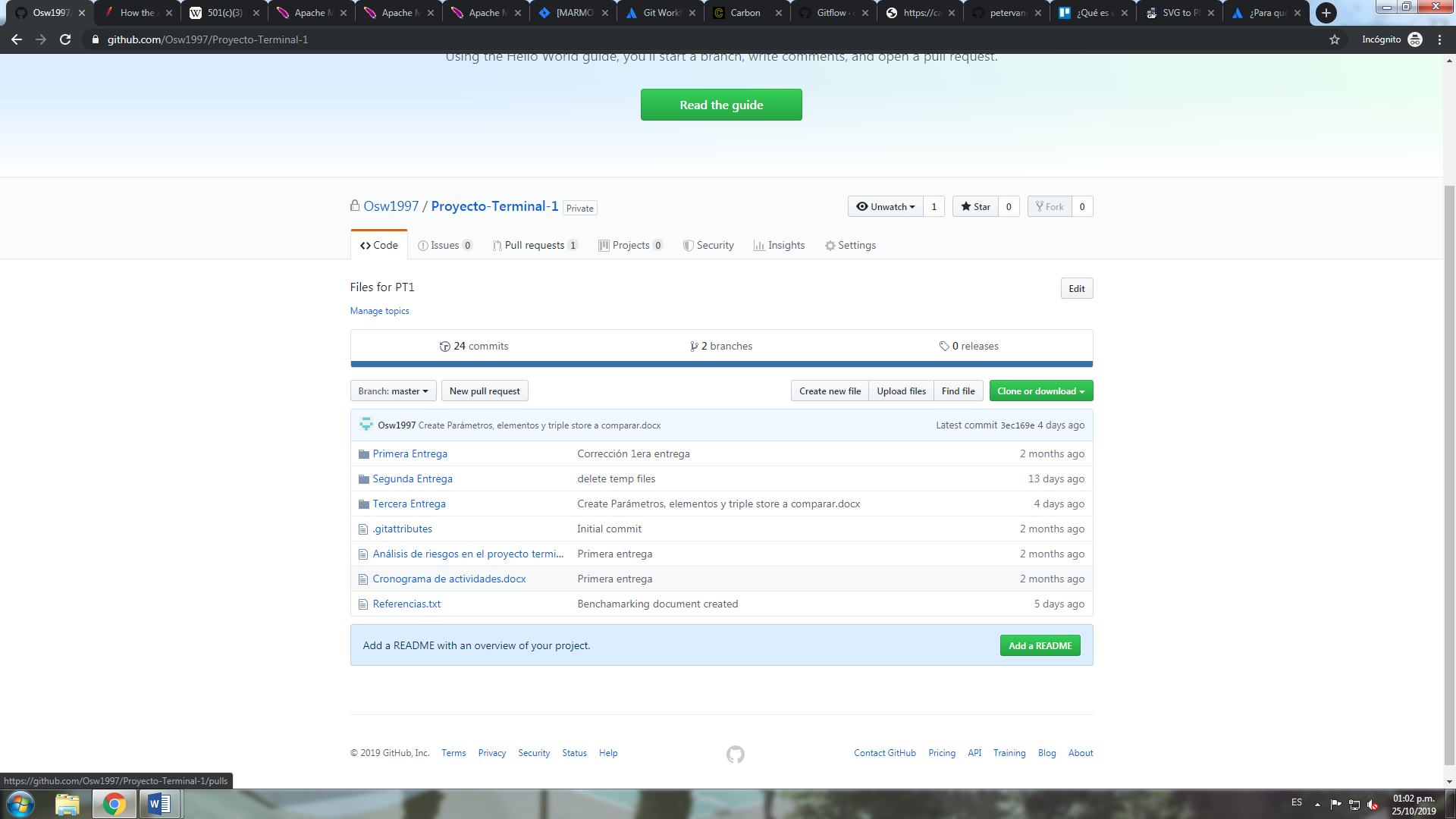
*Pull Request*

En la plataforma de control de versiones *Git*, cuando un desarrollador modifica código fuente de un cierto software con el fin de implementar una nueva característica, se lleva a cabo un *pull request* para que los desarrolladores oficiales del software validen el código modificado y determinar si se incorpora el código del ofrecido en el *pull request* o no en el código fuente oficial.

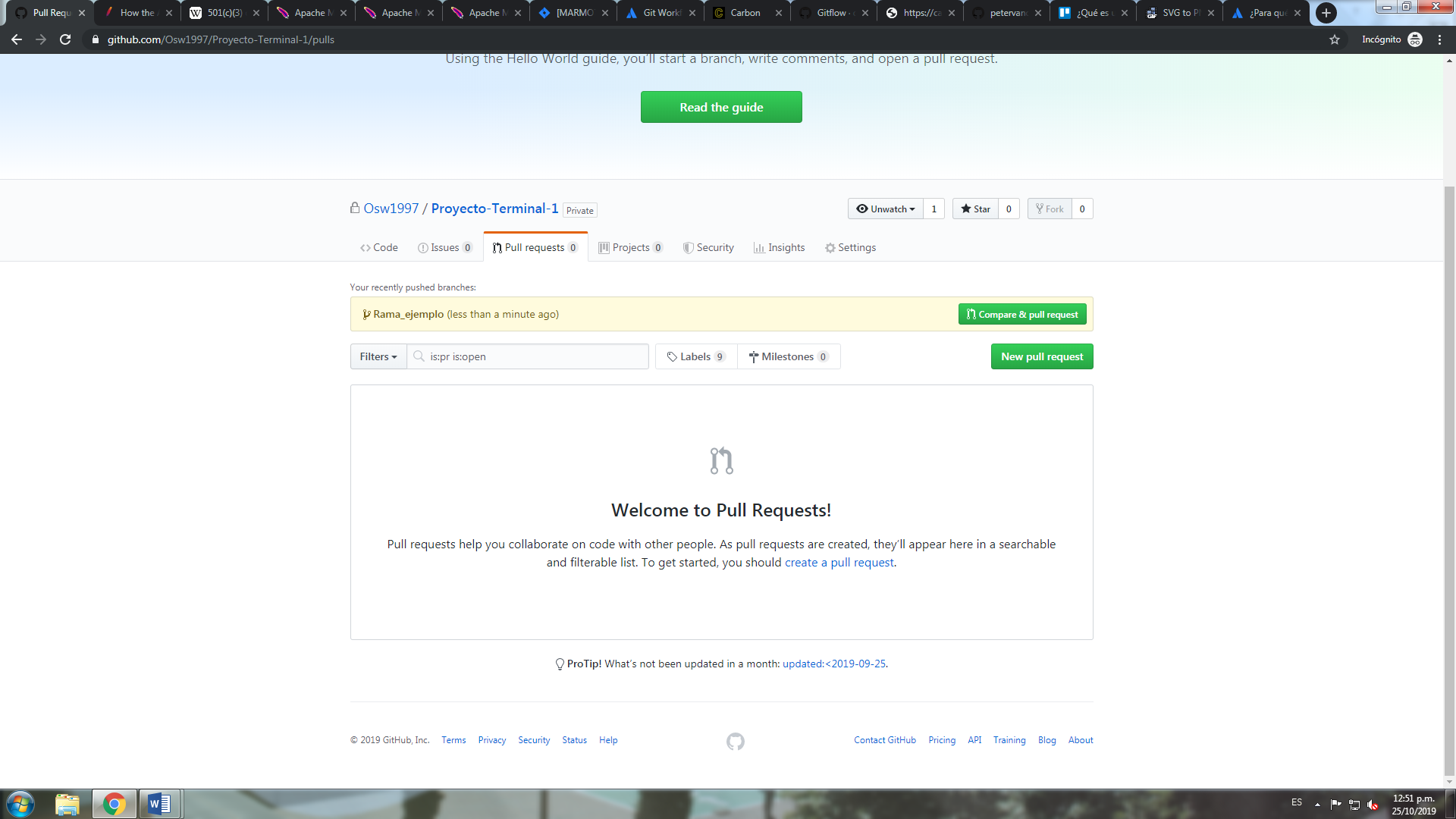
El primer paso es que en la respectiva cuenta de GitHub, dirigirse al repositorio donde se subió el código que se quiere contribuir.



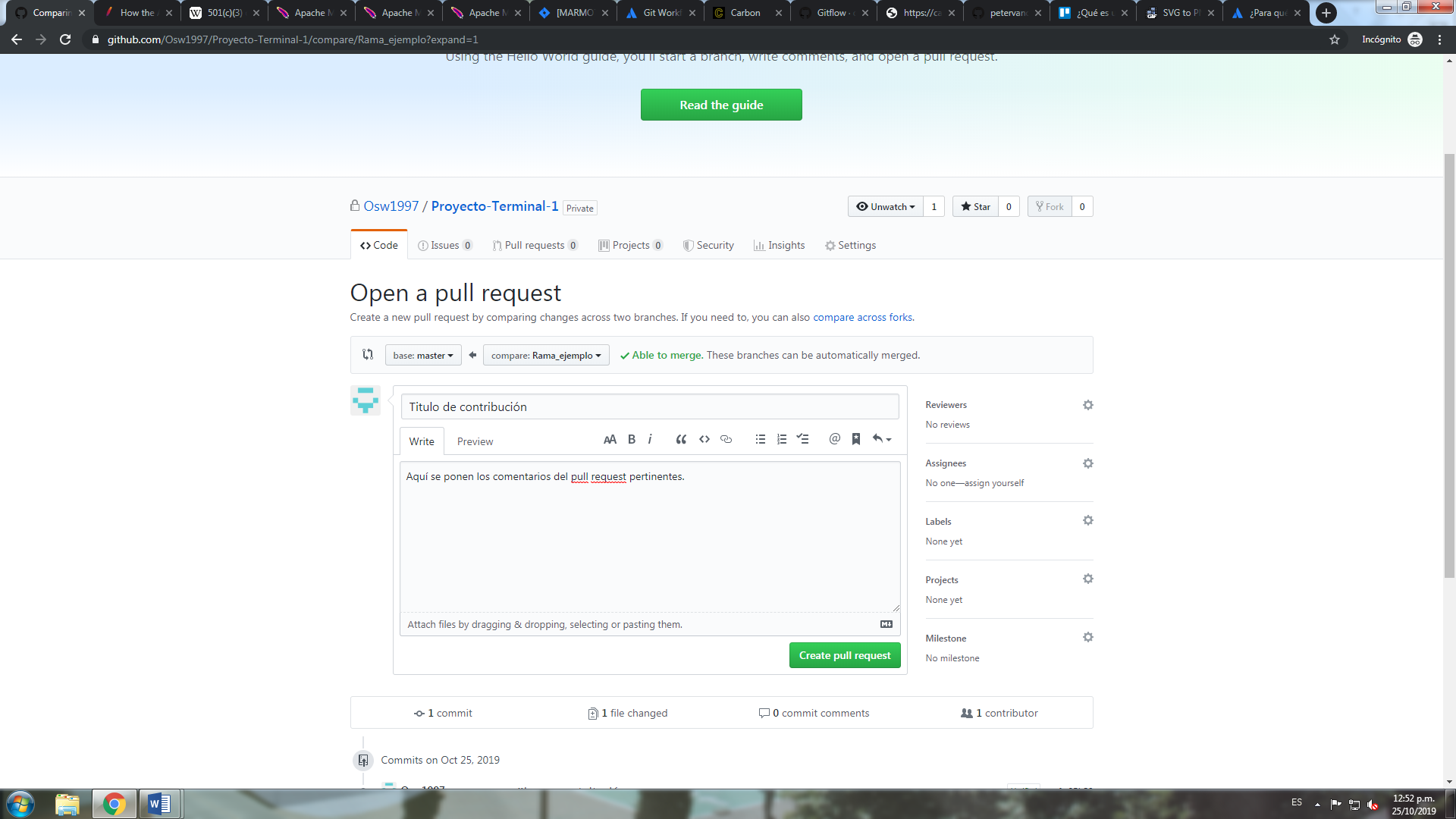
Después, se selecciona a la pestaña *Pull requests*



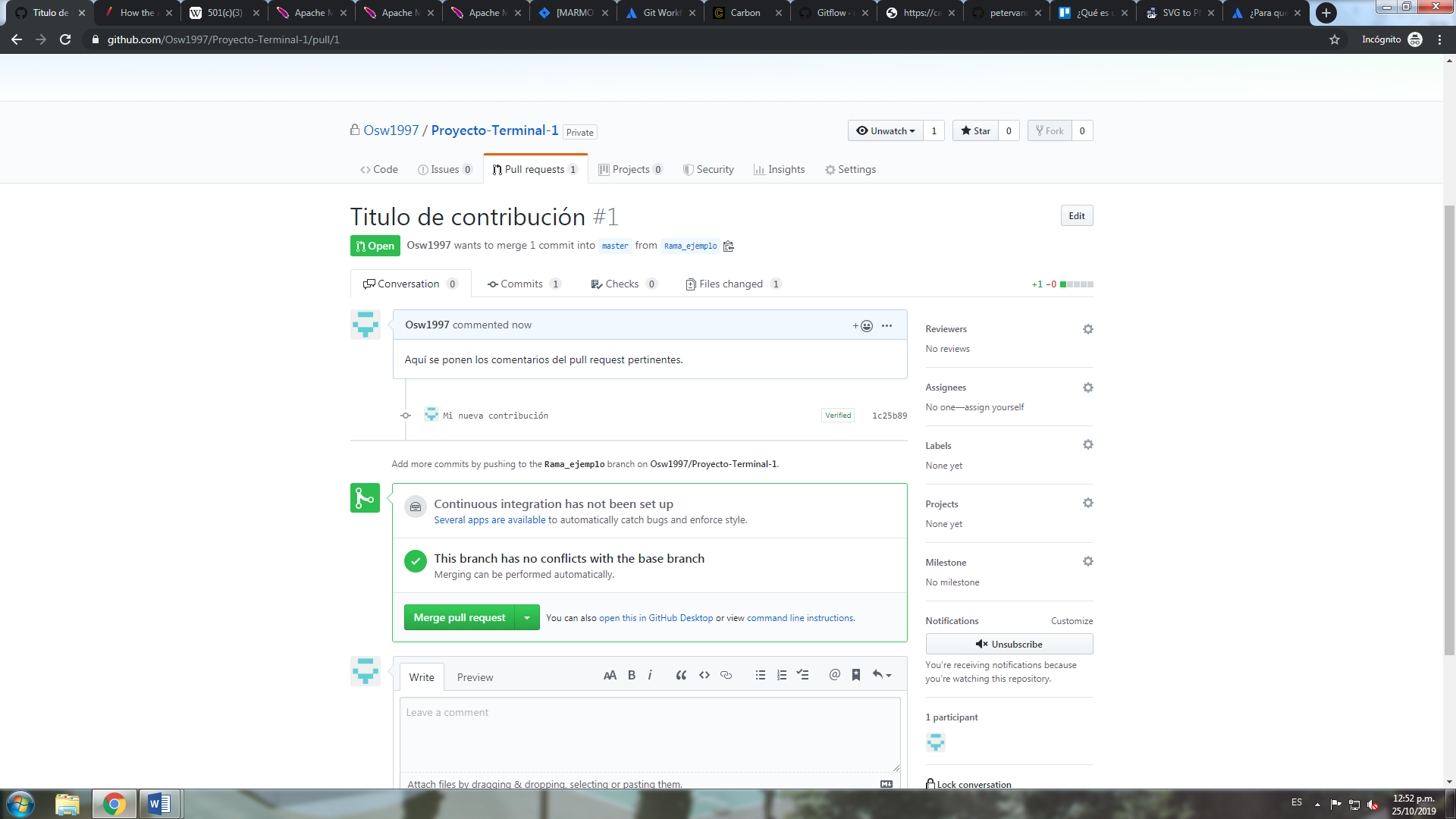
Luego, el código que se subió al repositorio debe de aparecer en un cuadro amarillo. Ahí mismo se debe de seleccionar el botón verde que dice *Compare & pull request*



Posteriormente se debe de escribir el título del *pull request*, agregar su descripción y seleccionar el botón verde *Create pull request.*



Finalmente, se creó un *pull request* sobre el repositorio tal y como se muestra en la figura. A partir de este paso, solo es cuestión de esperar a que los desarrolladores determinen si es aceptado o no el *pull request.*



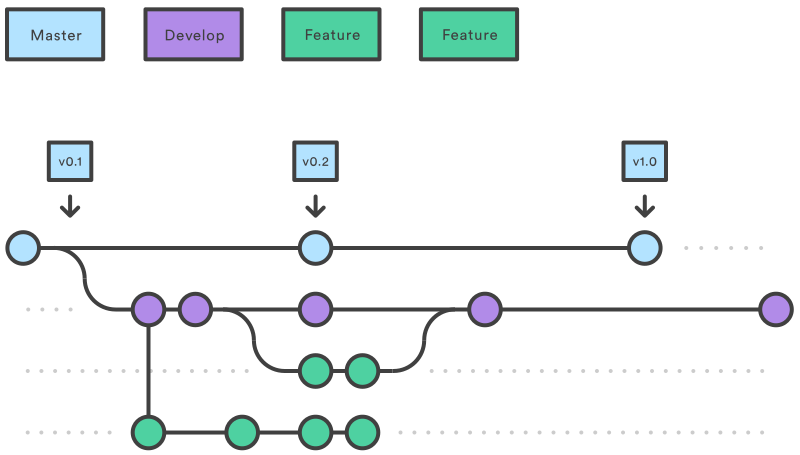
*Gitflow*

Es una secuencia de pasos para desarrollar un proyecto basado en *Git* publicado por Vincent Driessen. Este flujo de trabajo es un modelo basado en ramificaciones respecto a las versiones de un proyecto.

El objetivo de usar *GitFlow* es

* Colaboración
* Desarrollo en paralelo
* Soporte para corregir un error en el proyecto de manera rápida y eficaz.

Ejemplo de un diagrama sobre *GitFlow* es la figura 1



Jira software

Es una plataforma de la empresa *Atlassian* que permite gestionar el trabajo de un equipo de trabajo[Ref].

Lo que se puede gestionar en Jira son

* Requisitos
* Casos de prueba
* Desarrollo de software ágil

En *Jira* se proporciona

* Planificación de publicaciones y tareas
* Gestión de incidencias

*Apache Software Foundation (ASF)*

Es una organización sin fines de lucro fundada en 1999 apoyada económicamente por donaciones y patrocinadores corporativos. La *ASF* desarrollan proyectos *Open Source*, entre los cuales se encuentran el servidor Apache HTTP, Apache Kafka para *big data*, Apache *Spark* para aprendizaje de máquina por mencionar algunos.

Los objetivos de la *ASF* son:

* Crear una organización legalmente independiente de tal forma que individuos y compañías puedan aportar sus recursos y que sean usados en beneficio público.
* Proporcionar una organización colaborativa, libvre de desarrollo de software al proveer la infraestructura necesaria para que los proyectos sean realizados.
* Ser un recurso legal respecto a personas u organizaciones que contribuyan a los proyectos de la fundación.
* Proteger los proyectos de la fundación de abuso de otras organizaciones.

Apache Marmotta

Es un proyecto de la *ASF* cuya meta es ser una plataforma para *Linked Data* para ser usada, expandida e implementada por organizaciones que requieran administrar información *Linked Data* o crear sus propias aplicaciones en *Linked Data*.

Las características de Apache Marmotta son

* Servidor de escritura y lectura para datos del *Linked Data* cuya pila de tecnologías sean Java EE.
* Tripletas RDF basado en *Eclipse RDF4J*
* 7 tipos de transacciones ofrecidas por el *Kiwi triple store*.
* Consultas *SPARQL* y *LDpath.*
* Gestor de caché transparente para el *Linked Data Caching.*
* Ofrece mecanismos de seguridad básicos (Usuarios, perfiles y roles).

Apache Marmotta está basado en el protocolo *SPARLQ 1.1* y de las 11 funcionalidades que el protocolos menciona que toda plataforma de *Linked* *Data* debe poseer. Apache Marmotta implementa 9 de la 11, una de las cuales es el propósito del actual proyecto termina y es la capacidad de realizar consultas federadas.

Implementación del módulo de consultas federadas geoespaciales para Apache Marmotta

En cuanto a metodología y ciclo de vida del software, tal y como se mencionó en un trabajo pasado, será *Extreme Programming (XP)* debido a que las circunstancias del proyecto, es la metodología que más se adapta.

Por otra parte, cuando se tenga el producto terminado, se buscará aportarlo a la *ASF*. Para cumplir este objetivo, Apache Marmotta en su página oficial, ofrece una guía de cómo se debe realizar un *pull request* en el código fuente oficial.

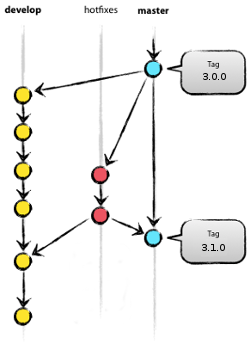
Apache Marmotta está basado en *Git* y en *GitFlow* pero con unas ligeras variaciones

* En vez de usar una sola rama *master*, se usan 2 ramas para mantener el registro del proyecto: *master master* y *develop*
  + La rama *master* es donde se aloja el código oficial de Marmotta.
  + La rama *develop* funciona para la inegrgación de características.

Las nuevas características pueden generar nuevas ramas que usualmente se identifican con la siguiente estructura: MARMOTA-XYZ, donde XYZ es un número que depende del *GitFlow.* Sobre esta rama es la que se estará trabajando para desarrollar el módulo de consultas geoespaciales.

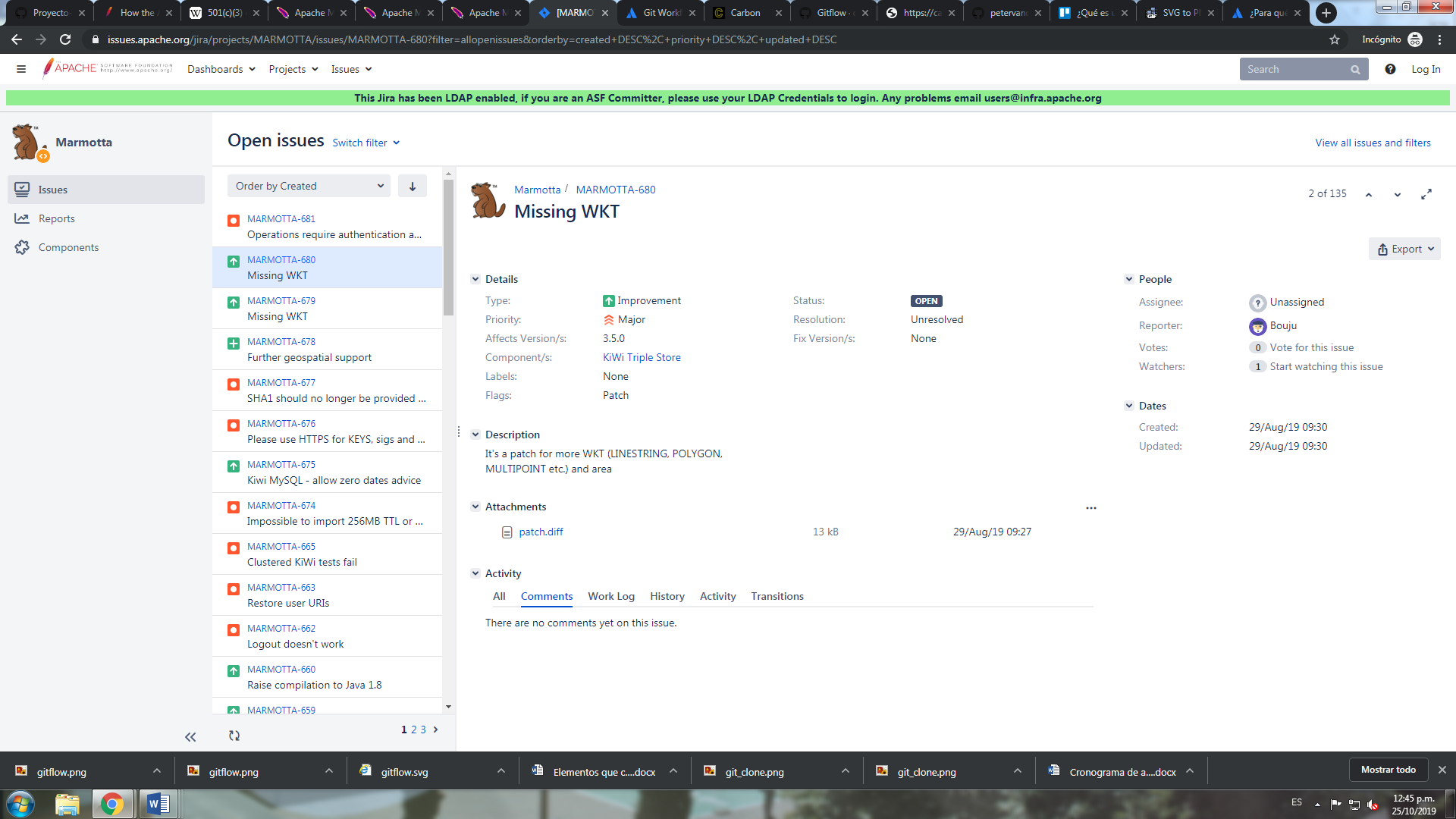
Para mantener el código se usa otra rama denominada *hotfixes* la cual sirve para corregir algún errro fácilmente del proyecto sin interrumpir el ciclo de vida del software. El identificador de esta rama es maintenance-3.0.X, donde X es número que dependerá también de flujo de trabajo. Probablemente también se tenga que trabajar sobre esta rama al estar desarrollando el módulo y se lleguen a presentar errores en alguna porción del código.

La figura 2 muestra cómo es que Apache Marmotta presenta su flujo de trabajo en *GitFlow*



Ya que todo el software de Apache Marmotta está alojado en *GitHub*, se puede hacer directamente el *pull request* con el fin de publicar la nueva característica o alguna corrección de código. Sin embargo, antes de hacer el *pull request* se debe de generar una publicación en el Jira de Marmotta^2 describiendo lo que se quiere implementar el el software. El título de la publicación también debe de ser el título del *pull request* con el fin de que el equipo de Mamotta pueda darle un seguimiento rápido y por ende, una respuesta al contribuidor que determine si se incorpora o no la contribución.

La figura 3 muestra el Jira de Apache Marmotta



Los pasos a seguir para llevar a cabo el desarrollo del módulo y después llevar a cabo el pull *request* son

* Clonar el repositorio oficial

git clone https://git-wip-us.apache.org/repos/asf/marmotta.git marmotta

cd marmotta/

git checkout MARMOTTA-584

* Desarrollar el módulo de consultas federadas con base a la metodología *XP*.
* Cuando se culmine el módulo, se debe de llevar a cabo el *pull request*.