

**Instituto Tecnológico Superior Zacatecas Occidente**

**Ingeniería en Sistemas Computacionales**



**“Manual Interno de Control de Versiones”**

**Gestión de Proyectos de Software**

**Profesora:**

Ing. Erika Jazmín Robles Gómez.

**Alumnas:**

Faviola Flores Salas

15040108

Itzel Damaris Salas Pérez

15040113

Cindy Lorena Domínguez Salazar

15040105

Karla Yaneli Conde Salazar

15040104

Brenda Marisol Figueroa García.

15040107

**11/Marzo/2019**

Manual de Control de Versiones

**Sistema de Control Escolar de Ingles SCEI**

**Estándar de Nombres**

Al crear cualquier documento con referencia al proyecto de inglés deberá tener una misma estructura misma que estará conformada por:

1. El número de versión como 0.1, 0.2 etc., según corresponda es decir si es la primera vez que el documento se sube al repositorio deberá contar con un 0.1 y en caso de requerir algún cambio o corrección deberá cambiarse la versión con un 0.2 y así sucesivamente.
2. Además de que el nombre no podrá contener conectores como de, a, y, o etc., por lo cual deberá solo agregar las palabras principales del documento con la primera mayúscula y unidos por un guion bajo “\_”.
3. Y para terminar se deberá agregar la fecha de la creación o modificación del documento comenzando por el día seguido del mes y terminando con el año esto unido con guiones medios justo como se muestra a continuación, Ejemplo:

**0.1Nombre\_Dearchivo\_01-03-2019**

**Nota:** Los archivos de checklist deberán también contener el nombre de la carpeta donde será guardado y además cada checlist deberá cambiar el número de versión en forma ascendente. Ejemplo:

**0.1Nombre\_Dearchivo\_** **Checklist\_Ingles \_01-03-2019**

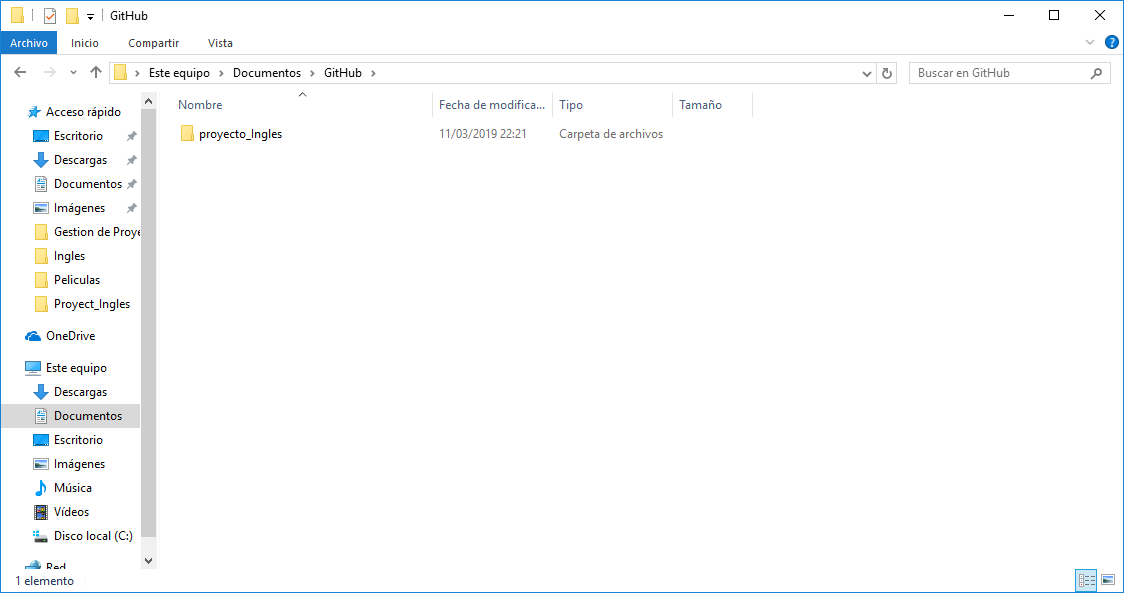
**0.2Nombre\_Dearchivo\_** **Checklist\_Ingles \_01-03-2019**

**0.3Nombre\_Dearchivo\_** **Checklist\_Ingles \_01-03-2019**

**Estructura de carpetas**

Para contar con un buen manejo de la información dentro de Git Hub debemos realizar una estructura de carpetas ordenadas de una forma que sea fácil de comprender y de utilizar para todos los integrantes del equipo ya todos los archivos derivados del proyecto\_Ingles serán respaldados en este repositorio.

1. Dentro del repositorio de Git Hub será creada la carpeta del proyecto la cual tendrá por nombre proyecto\_Ingles.



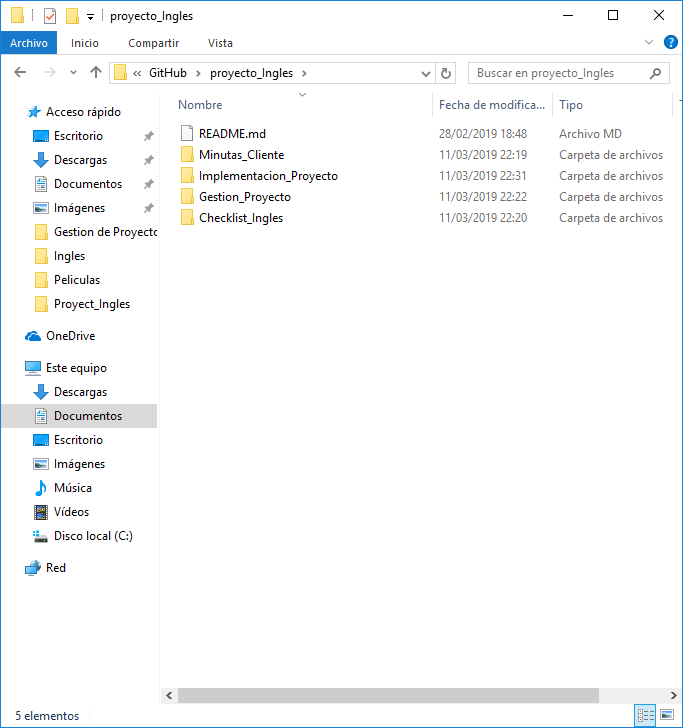
1. Dentro de proyecto\_Ingles se Contendrán las carpetas:

-Minutas\_Cliente

-Implementacion\_Proyecto

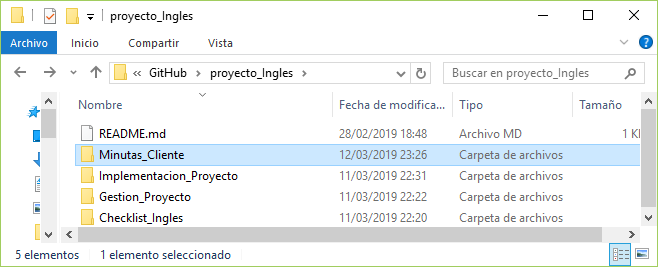
-Gestion\_Proyecto

-Checklist\_Ingles



1. A continuación, se explicará que documentos se guardaran en cada carpeta.

Dentro de Minutas Cliente se deberán agregar las minutas generadas a lo largo del proyecto.



1. Dentro de Implementacion\_Proyecto se encuentra otra serie de carpetas

-Análisis y Requerimientos:

Dentro de esta carpeta se deberá guardar el archivo de SRS, el SAS, el Diseño de interfaz, el historial de usuario y el diccionario de datos en caso de tener uno.

-Construcción:

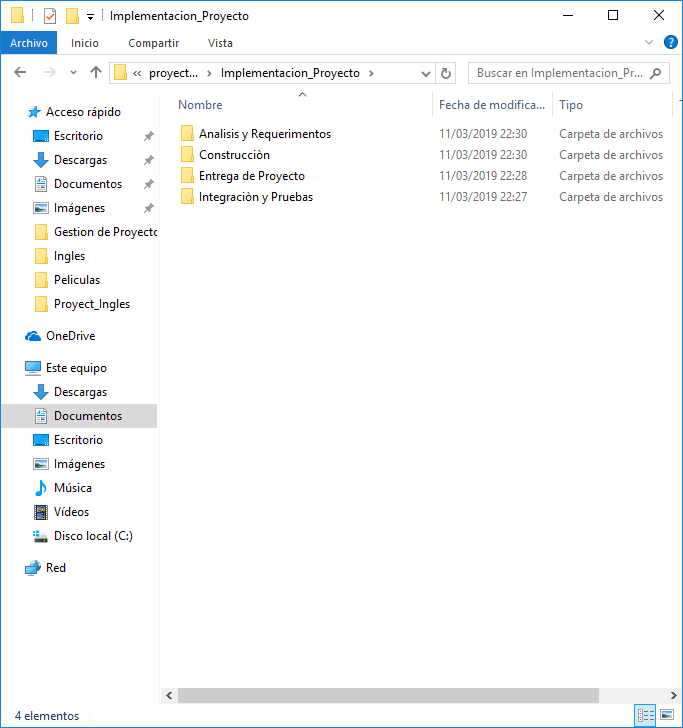
Aquí se alojará todo el código generado.

-Entrega de Proyecto:

Acta de cierre, Manuales de Usuario, Manual de Mantenimiento y Manual de Operación.

-Integración y Pruebas:

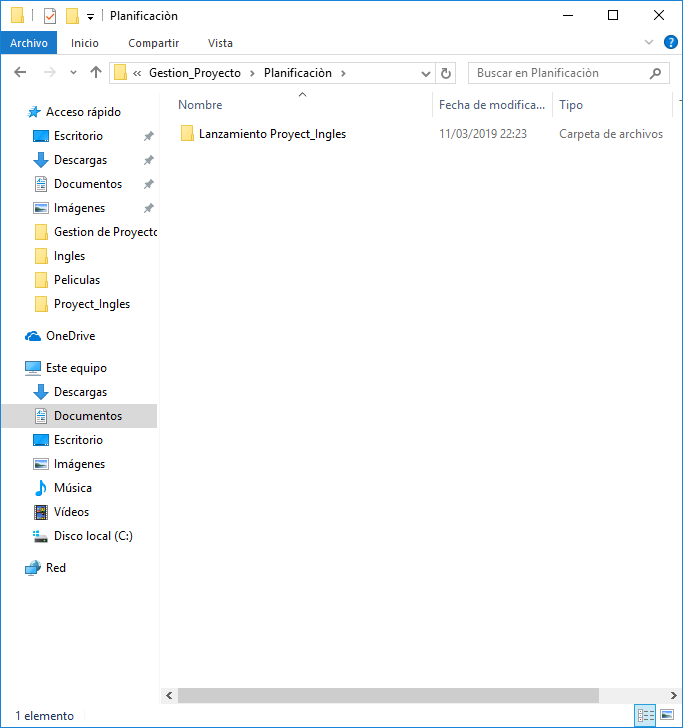
Aquí se guardaran todos los documentos que tengan referencia a pruebas (Unitarias, de integración. ETC.).



1. Dentro de Gestion\_Proyecto se encuentra:

-Planificación:

Aquí se encuentra una carpeta con el nombre de Lanzamiento Proyect\_Ingles aquí se deberá contener todos los documentos de la reunión de lanzamiento (Plan de Calidad, Plan del Proyecto, Seguimiento objetivos, Seguimiento Riesgos, Minuta reunión de Lanzamiento, Plan de adquisición, ETC.).

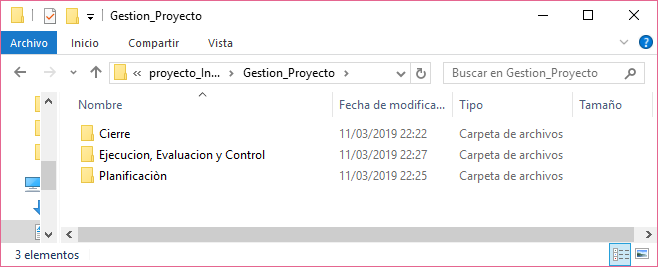


-Ejecución, Evaluación y Control:

Todas las reuniones de estatus y reuniones de avance con el clente.

-Cierre:

Acta de cierre de proyecto.

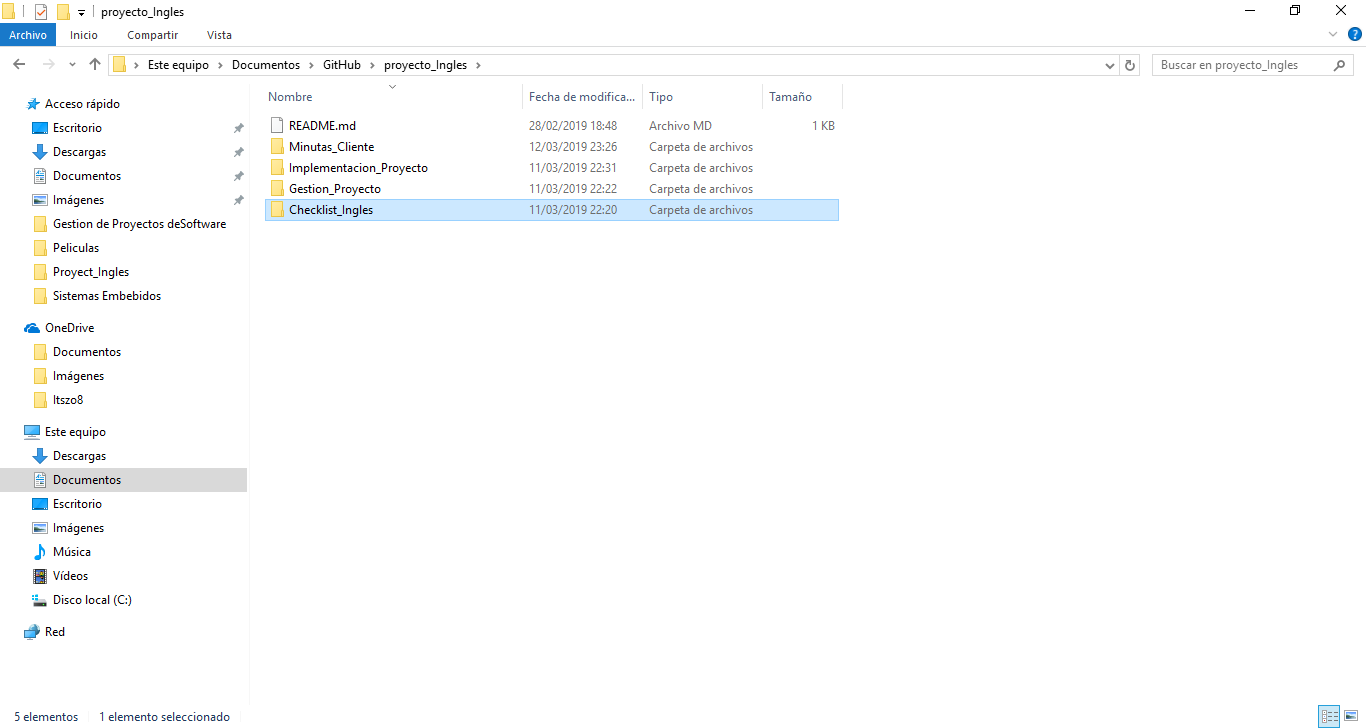


1. Dentro de la carpeta Checklist\_Ingles está carpeta cuenta con su propio estándar de nombres el cual se encuentra explícito al principio de este documento: verificación2

**0.1Nombre\_Dearchivo\_** **Checklist\_Ingles \_01-03-2019**

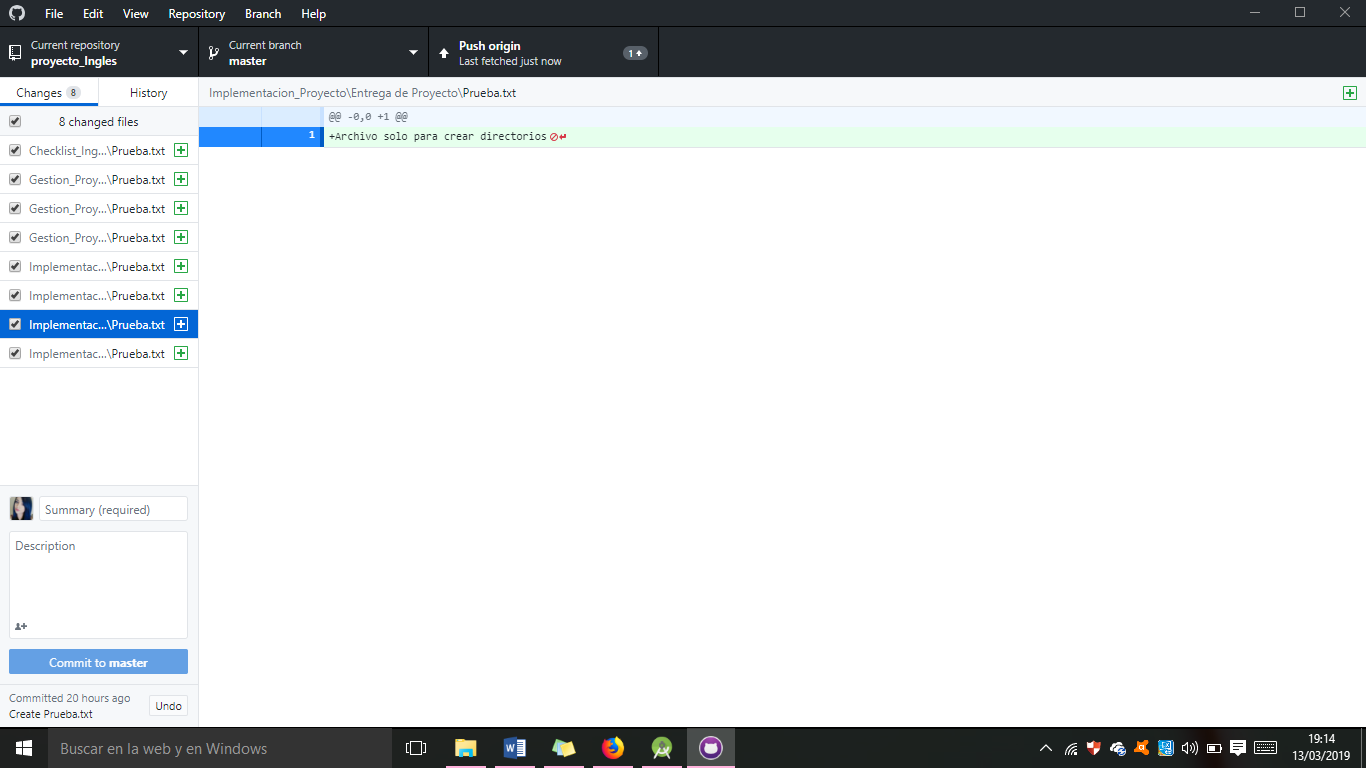
**0.2Nombre\_Dearchivo\_ Checklist\_Ingles \_01-03-2019**

**0.3Nombre\_Dearchivo\_ Checklist\_Ingles \_01-03-2019**

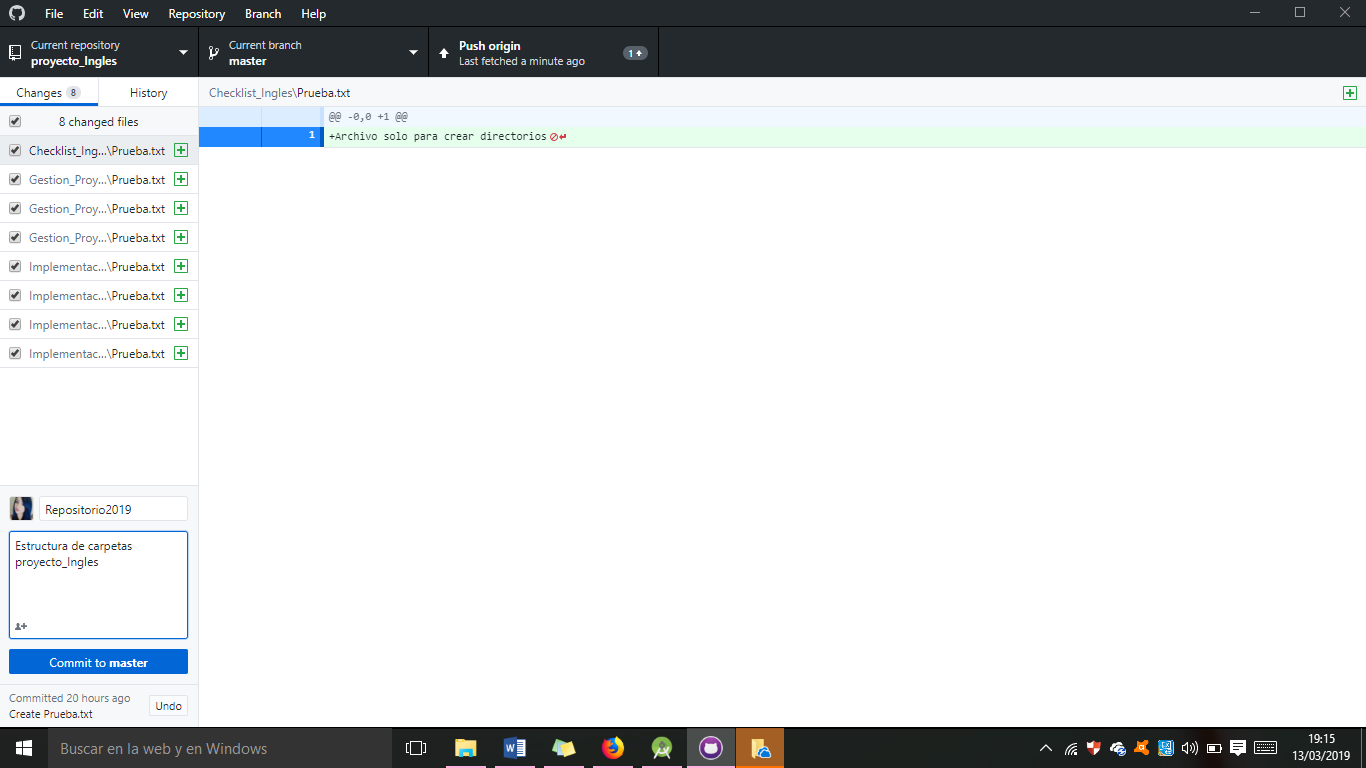


**Instrucciones para guardar archivos en el repositorio**

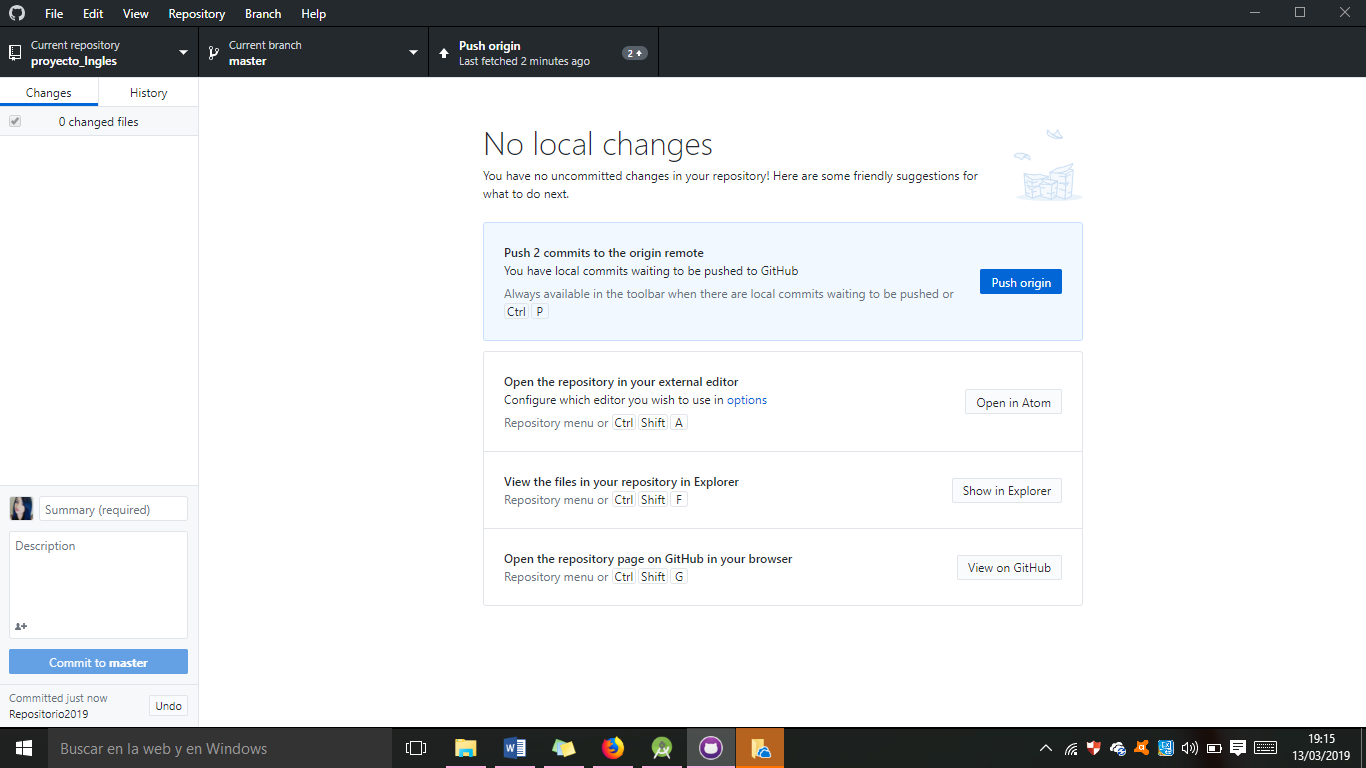
1. Manual para administrador de Git Para poder subir cualquier archivo al repositorio de Git Hub es necesario primero realizar una estructura ordenada de carpetas y a su vez incluir dentro de ellas un archivo de muestra para poder subir las carpetas.



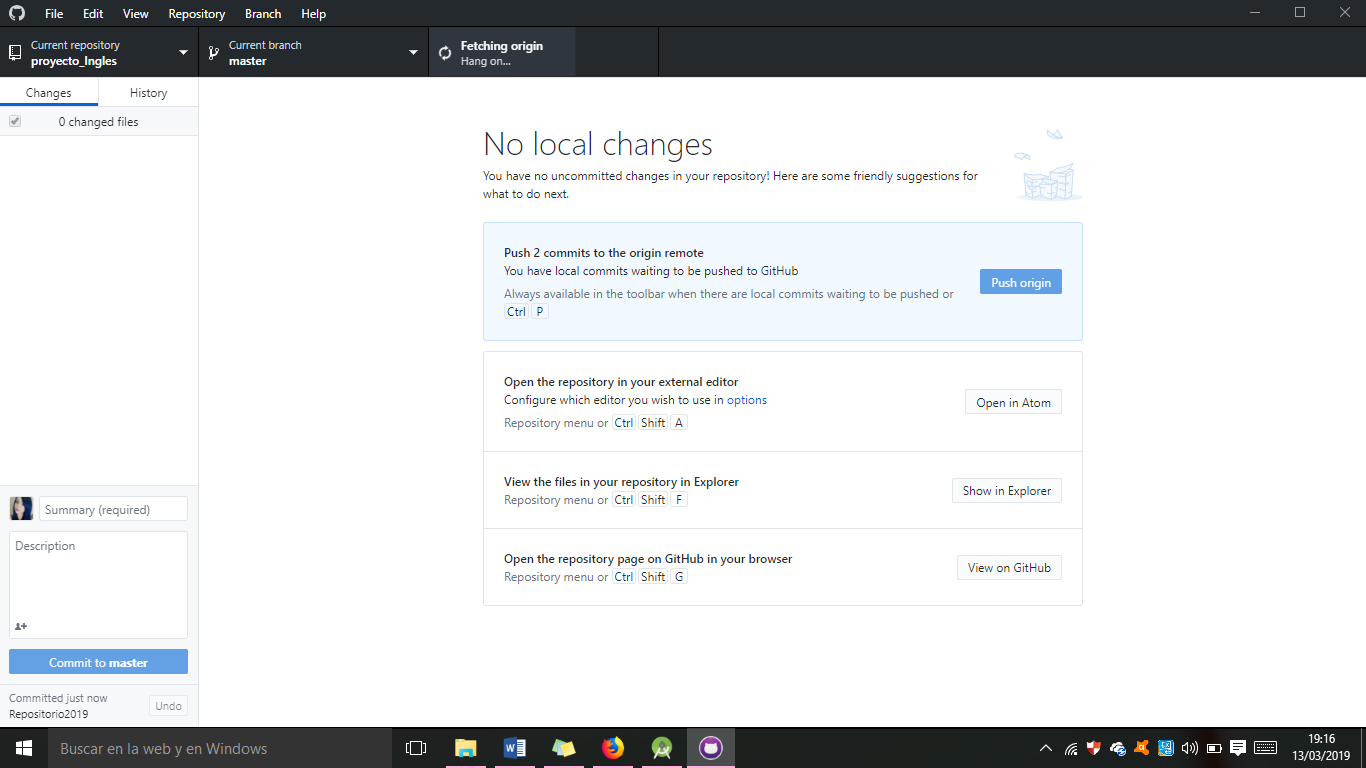
Es necesario saber que si no se hace un summary y un commit no se podrán subir los archivos en estos se coloca un sub-nombre o una palabra clave y en el commit es una pequeña descripción del archivo.



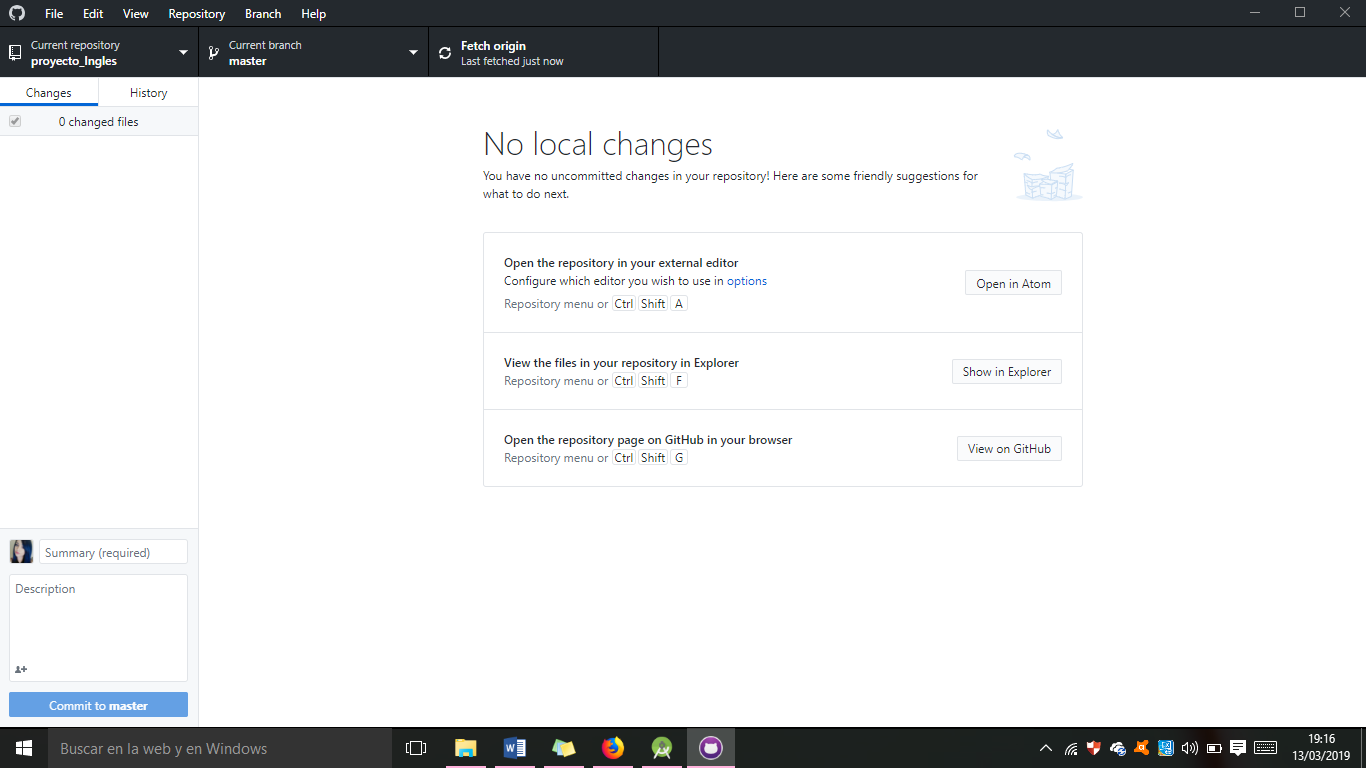
Una vez creado el commit se selecciona el Push origin.

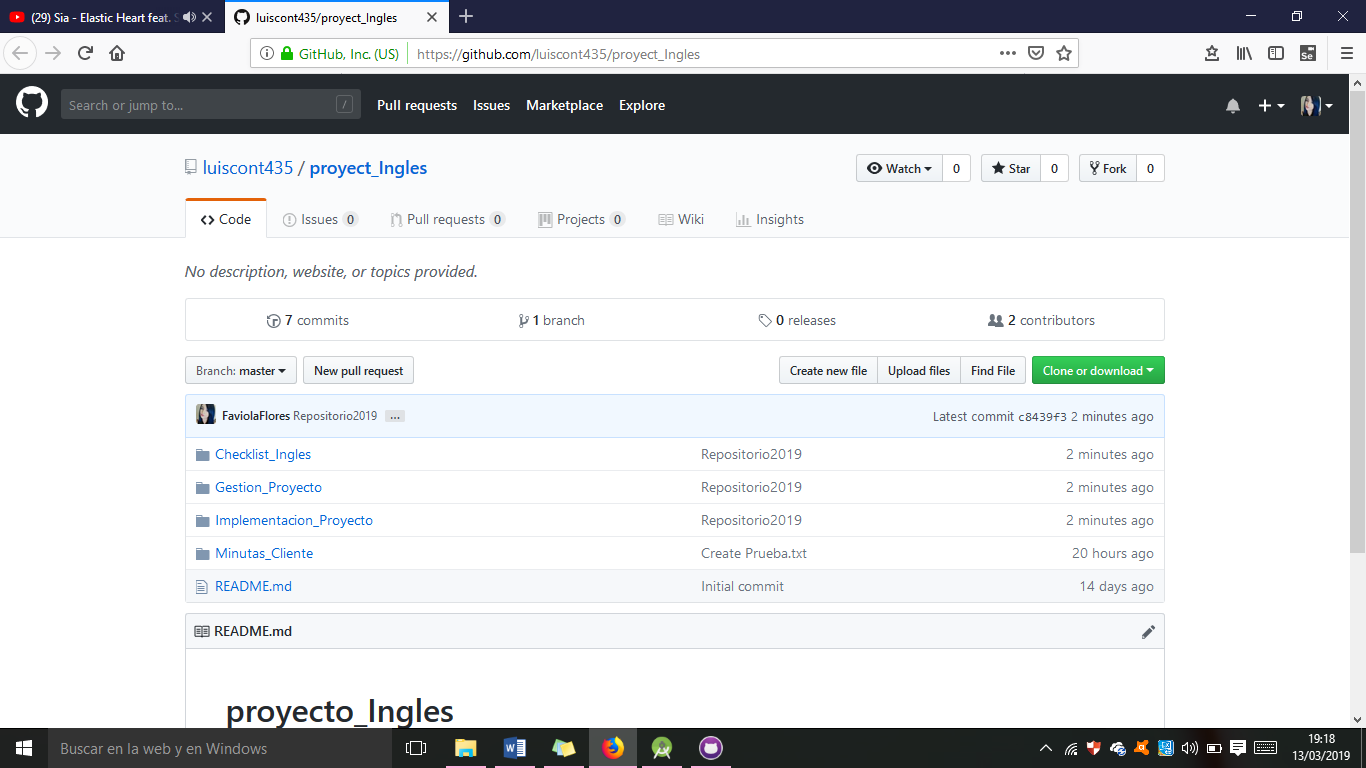


A continuación, se puede observar que ya se están subiendo los archivos comentados.



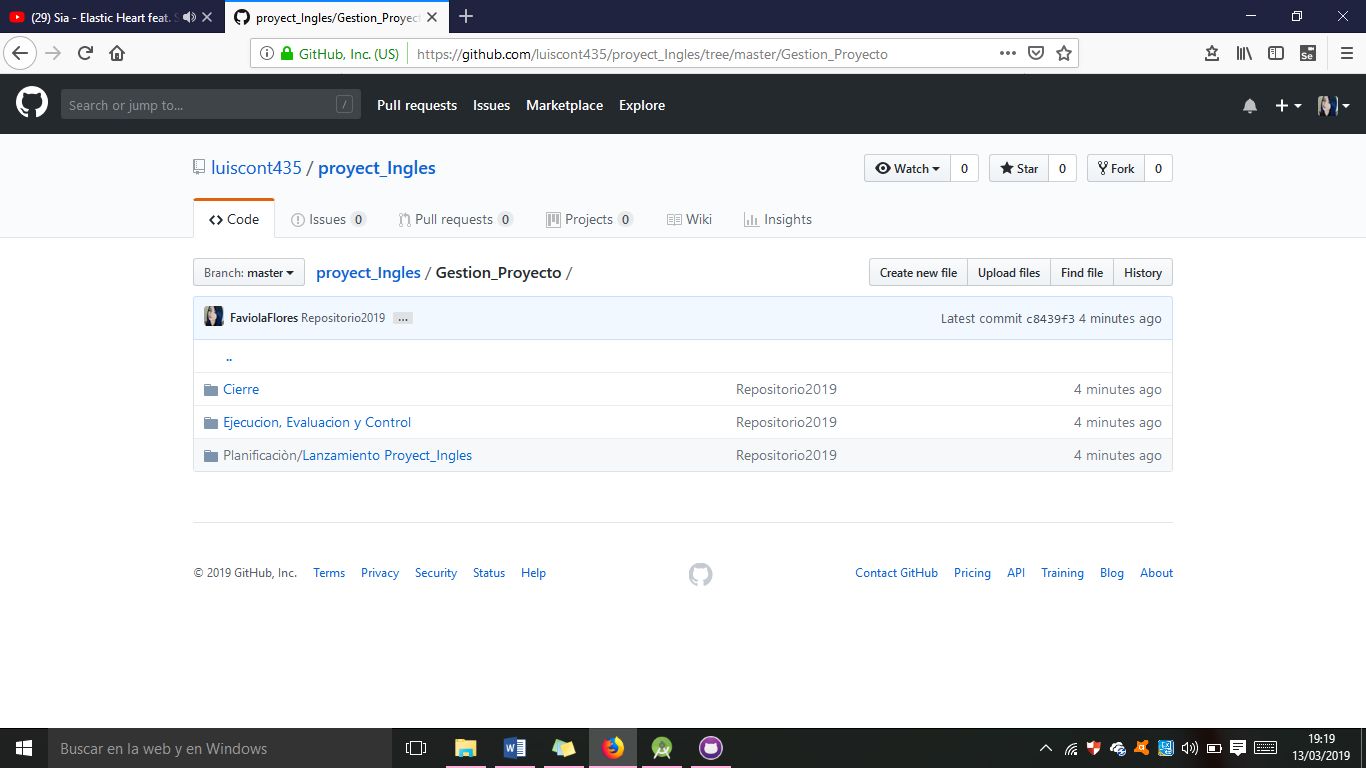
Cuando se termiana de cargar nos aparece la opción de ver en GitHub.



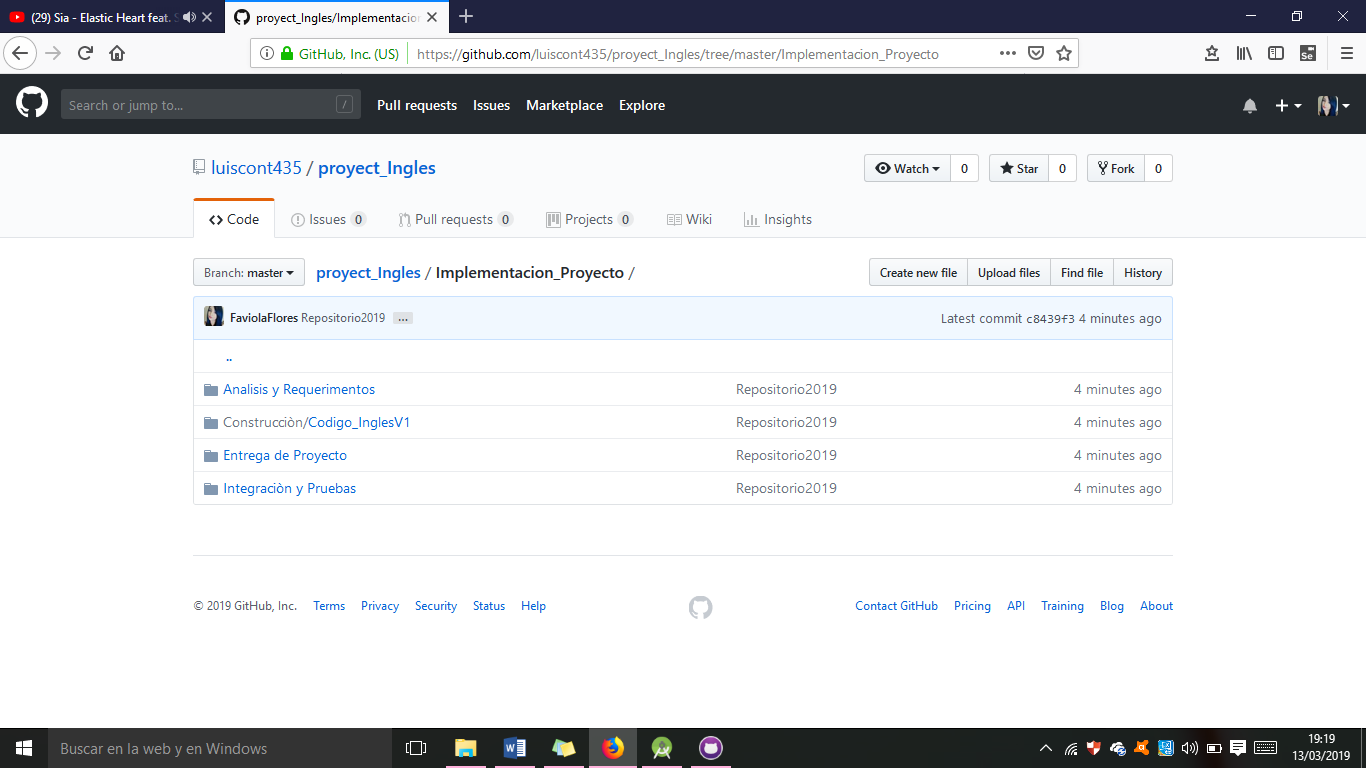
Y una vez dentro de Git ya aparecerán los archivos subidos acompañados del commit que se agregó.

Todos ellos con sus respectivas carpetas asignadas.

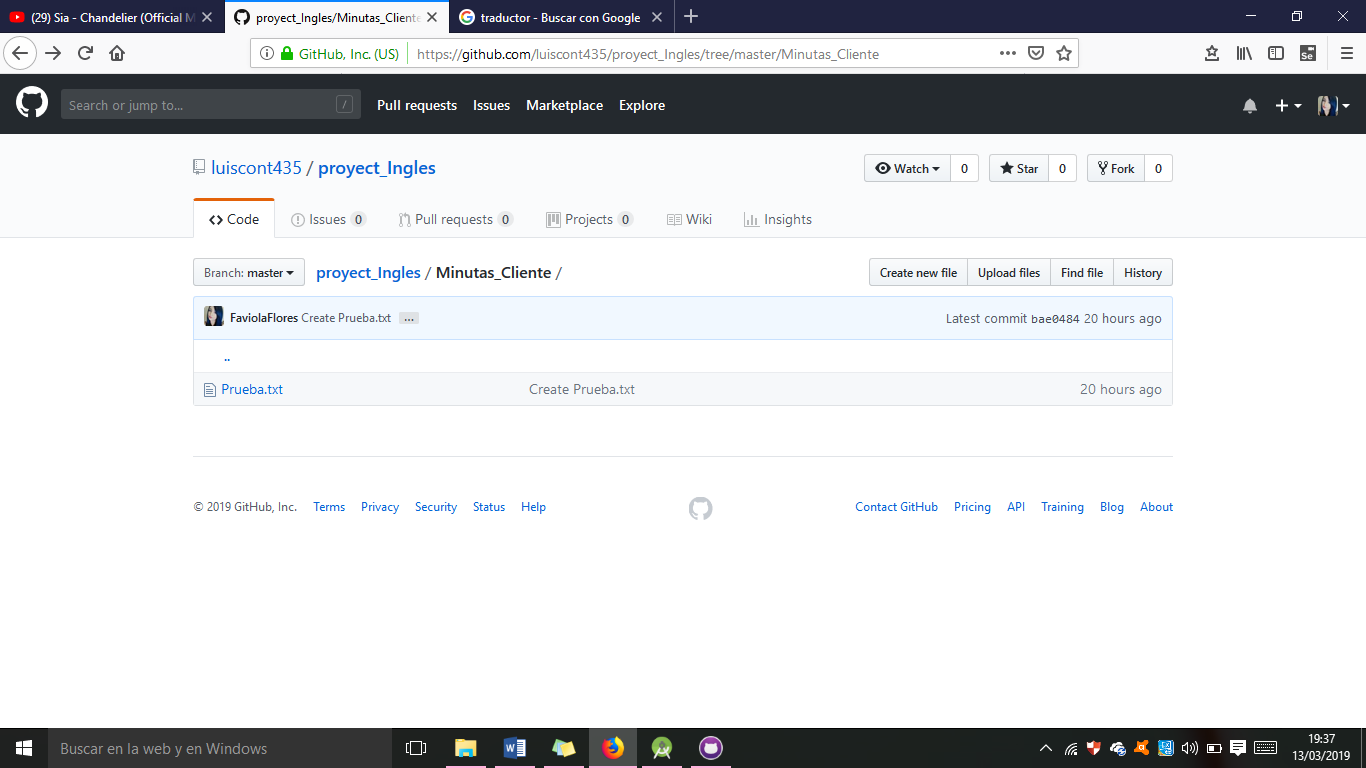
Gestion\_Proyecto



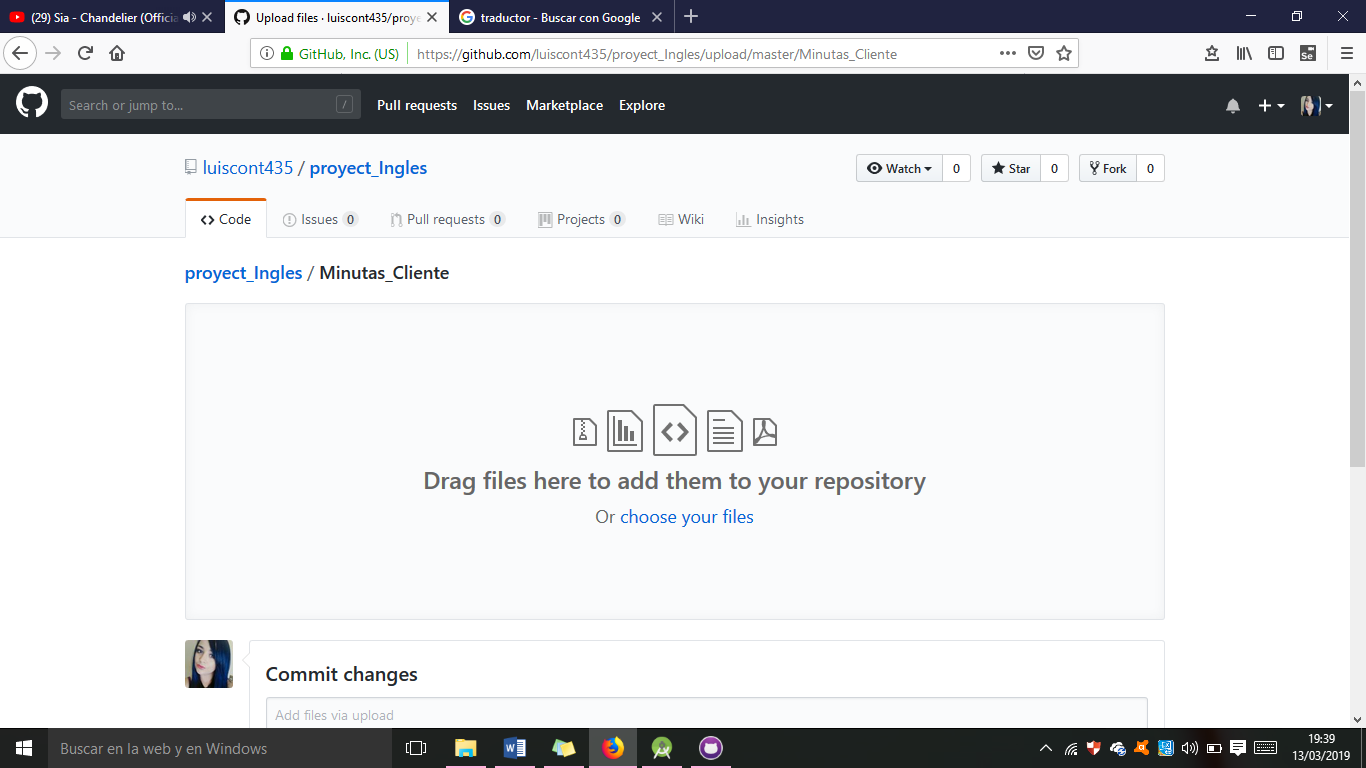
**Implementacion\_Proyecto**



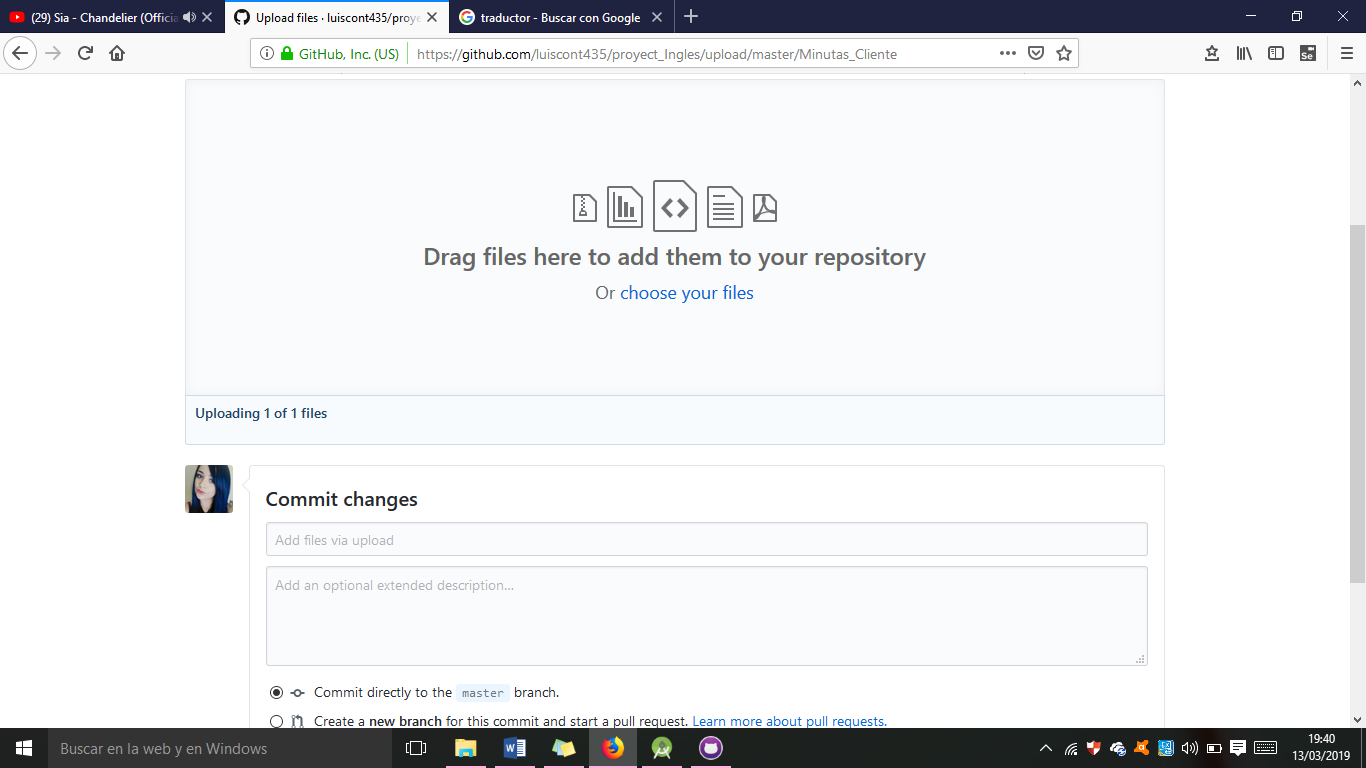
Con el fin de realizar un ejemplo en la subida de archivos al repositorio seleccionaremos la carpeta de Minutas\_Cliente. En seguida seleccionamos la opción de subir archivo.

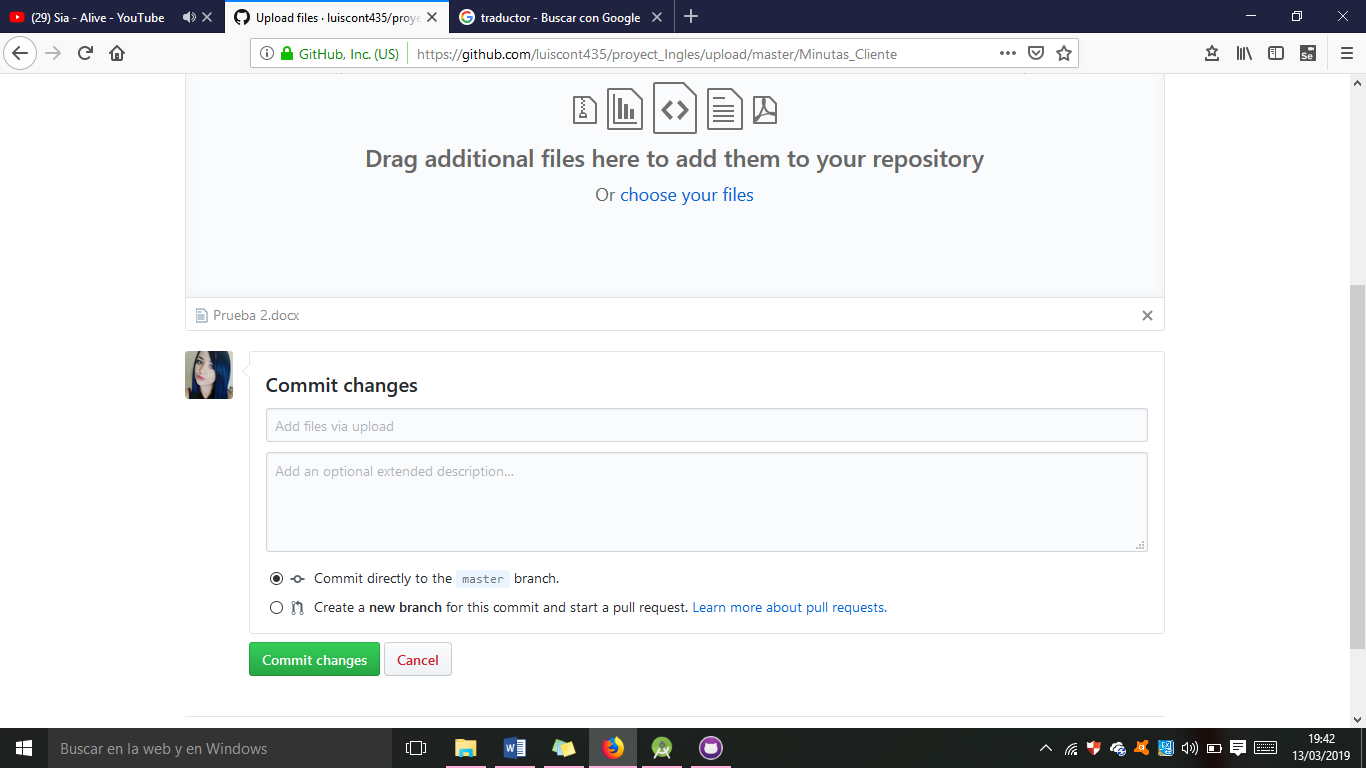


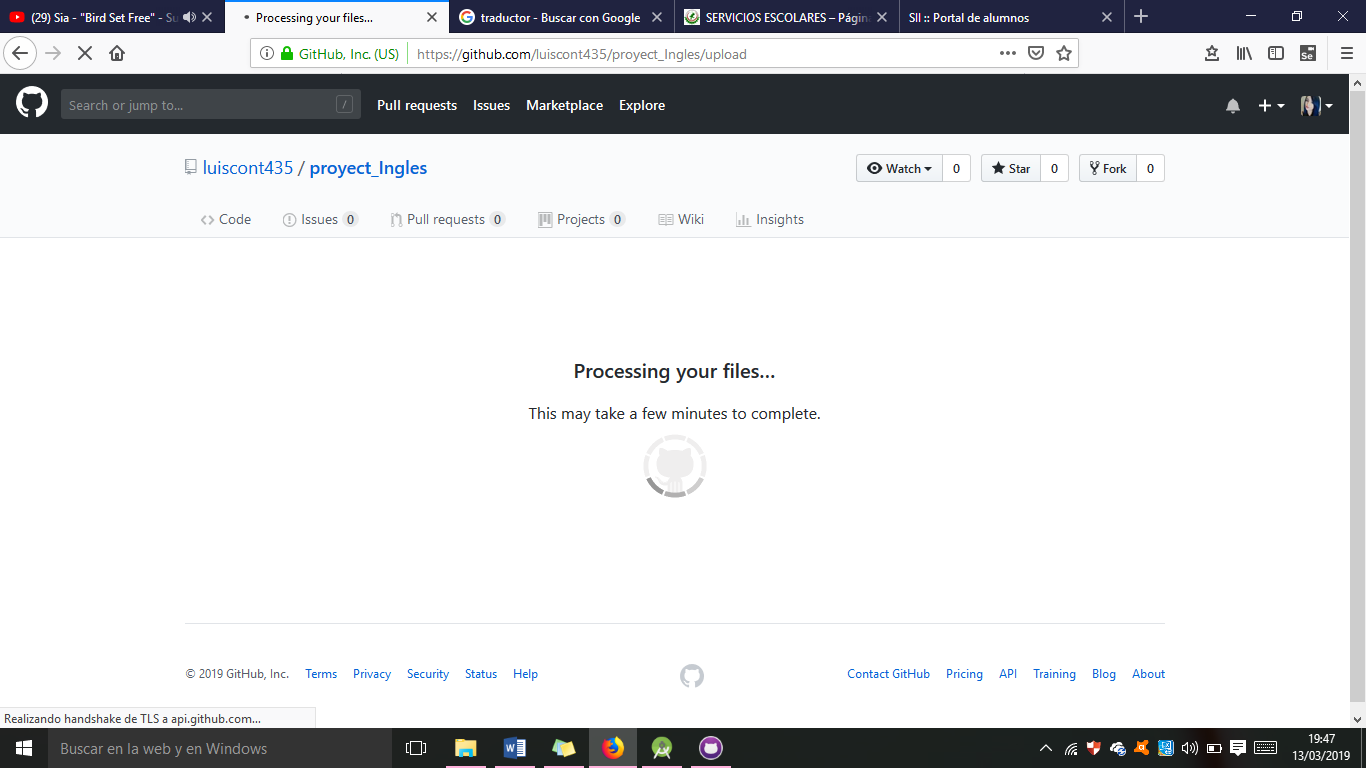
Y a partir de ahí se mostrará una pantalla como la siguiente donde podrás arrastrar documentos de cualquier tipo, ya sea un archivo PDF, Word, Exel, Zip etc.

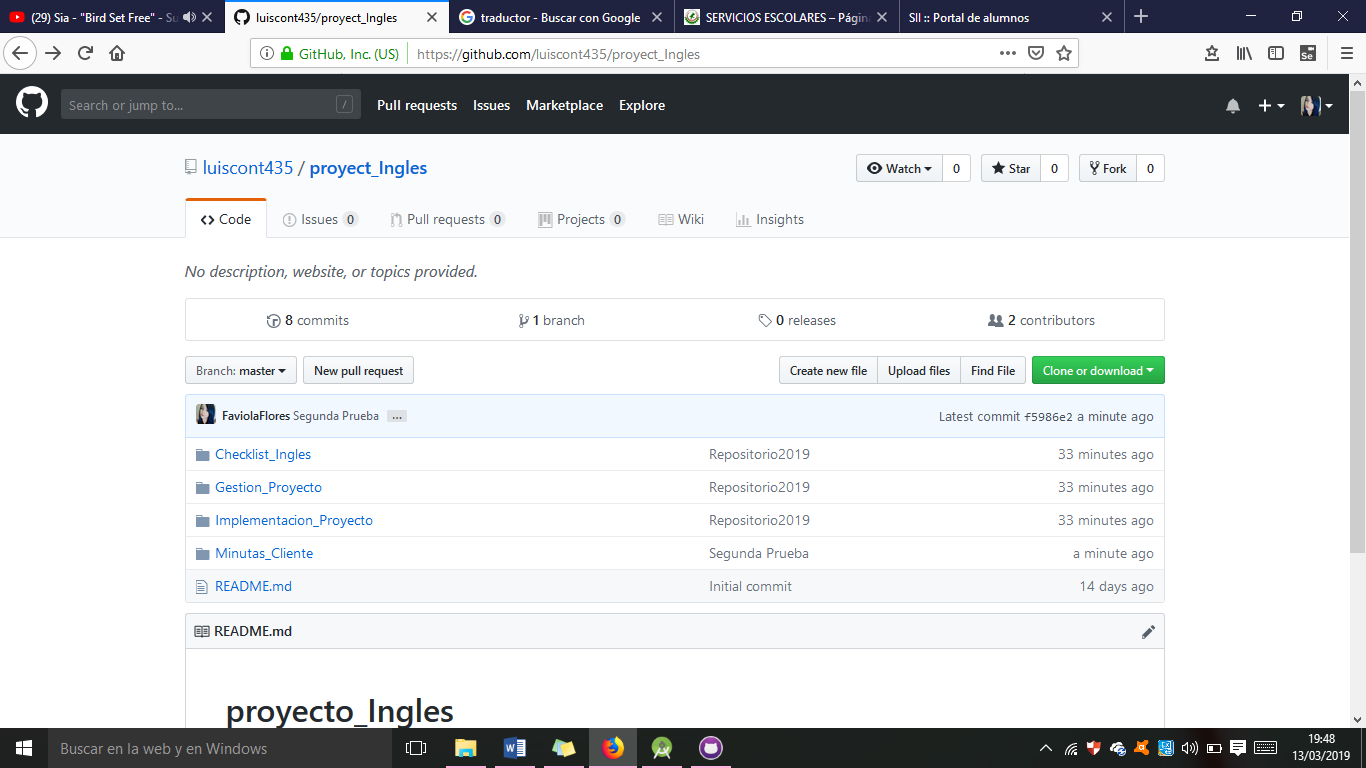


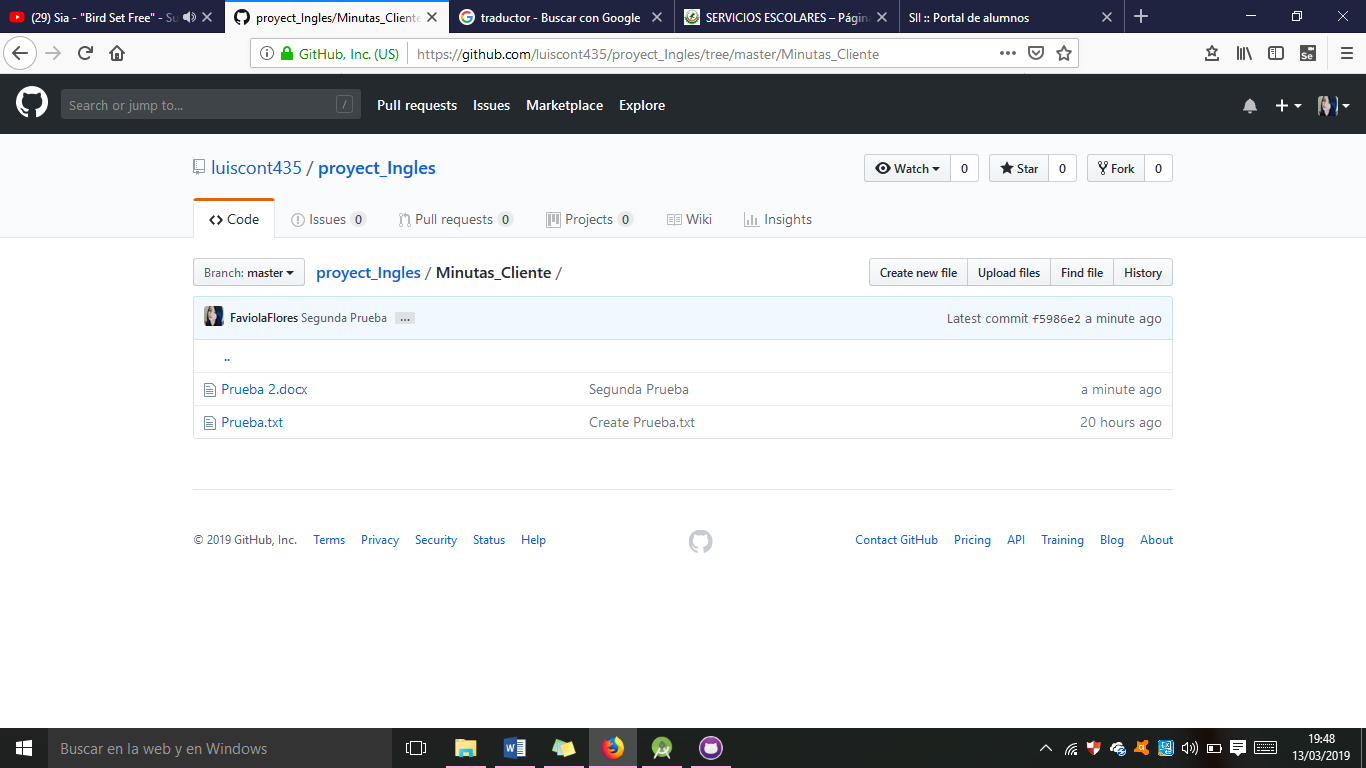
A continuación, se agregan los commits ya que sin esta parte llenada no se podrá realizar el cambio.



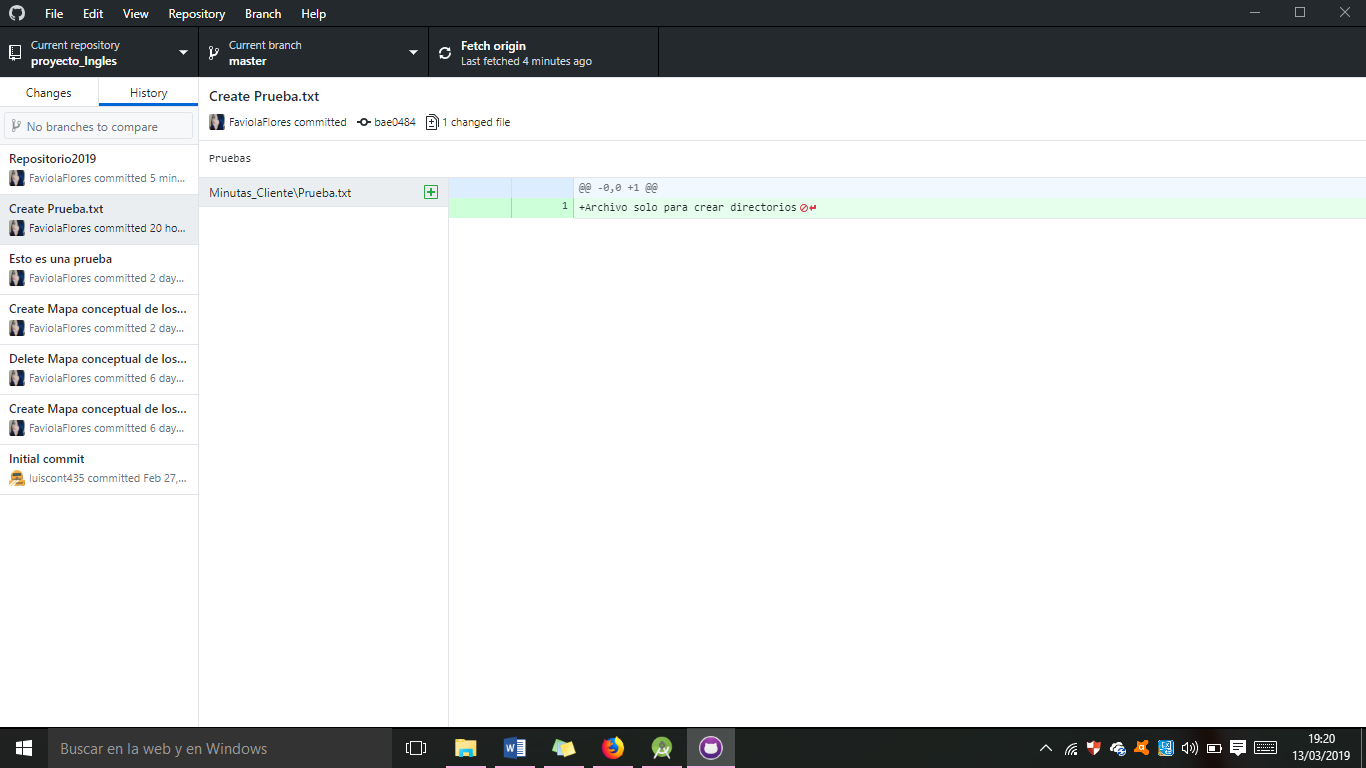








También contamos con un historial que nos ayudara a saber los cambios que se han realizado.



Con esto concluimos con el manual de versiones de SCEI.