**DOCUMENTACIÓN DAPW**

Índice

[1 Instalación en local 2](#__RefHeading___Toc107_3047387408)

[2 Hosting elegido 2](#__RefHeading___Toc109_3047387408)

[3 SEO OnPage 2](#__RefHeading___Toc111_3047387408)

[4 Control de versiones 2](#__RefHeading___Toc113_3047387408)

[4.1 El proceso para hacer commit es el siguiente: 2](#__RefHeading___Toc115_3047387408)

[4.2 Registro de commits 4](#__RefHeading___Toc117_3047387408)

# Instalación en local

# Hosting elegido

He seleccionado este servidor hosting entre otros por que tienes discos SSD Nvme, los servidores están conectados a una red de 1GB/s, tiene un ancho de banda ilimitado. Estos servidores tienen protección antiddos, tilera y arbor. Además, están ubicados en Europa, tienen acceso a SSH y usan las versión de PHP 7.4.

# SEO OnPage

# Control de versiones

URL repositorio GitHub: <https://www.github.com/ajsc2001/peluqueriaUnisexLaGallega>

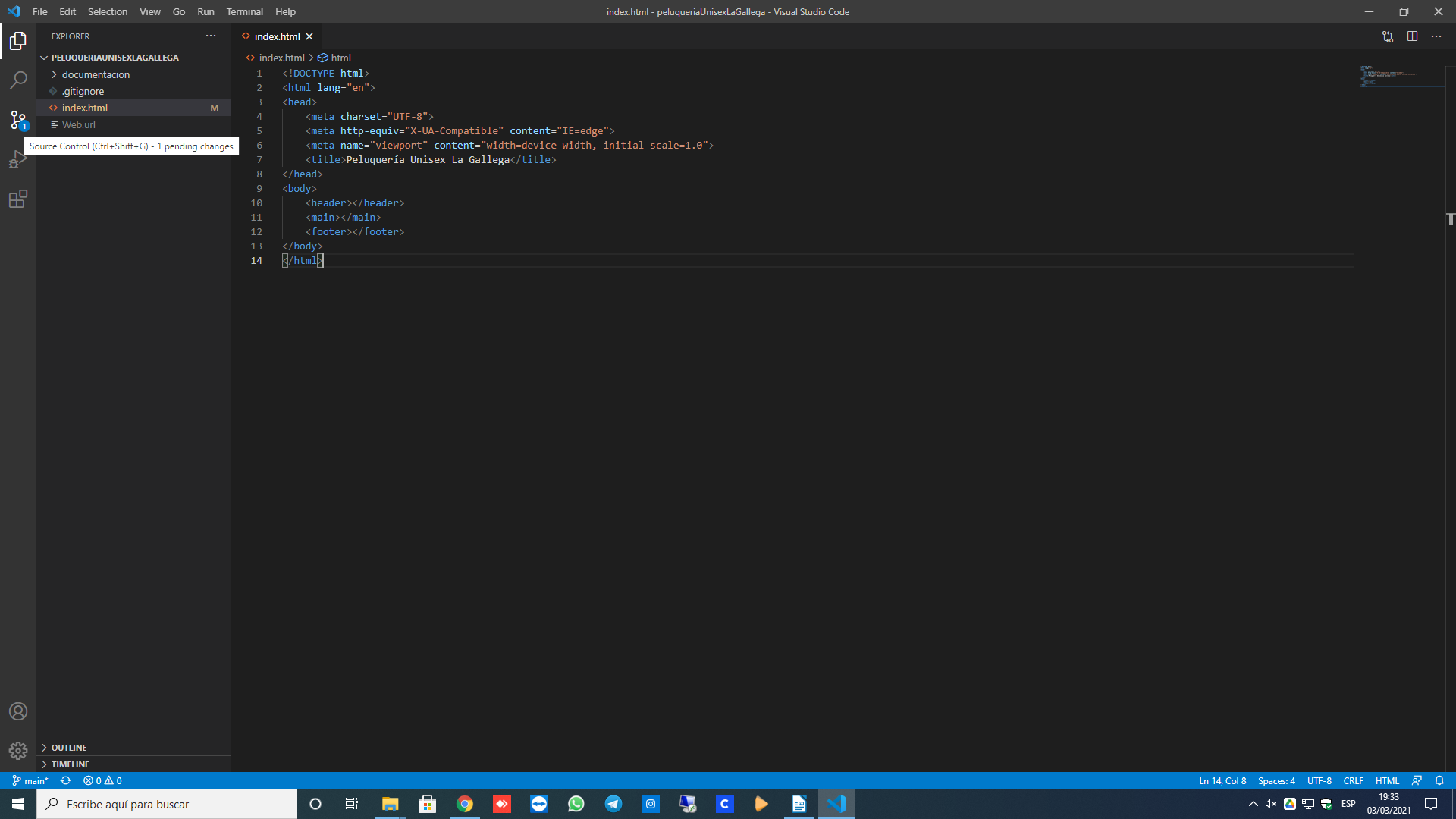
He creado este repositiorio vacio. Hago un git clone del repositorio y despues empiezo a hacer commits. Después he generado archivos como .gitignore y algunos otros borrado su contenido y he empezado a añadirle las reglas.

A la hora de hacer los commit en vez de usar la terminal de Git con los comandos: “git add .” y “git commit -m ‘Mensaje del commit’”. He usado la interfaz grafica que me ofrece Visual Studio Code.

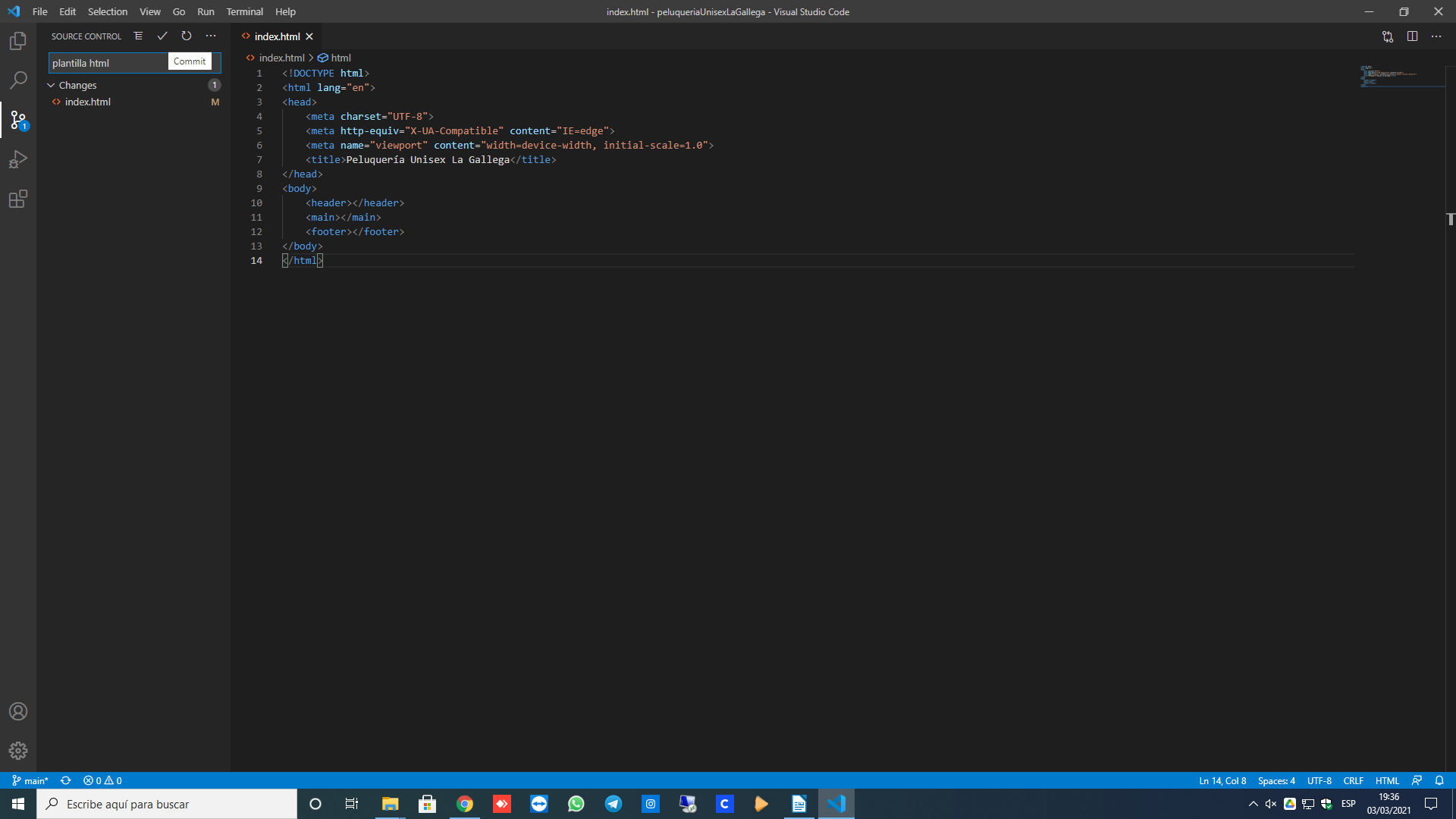
## El proceso para hacer commit es el siguiente:

Si se revisan todos los commits que tiene este repositorio no encontrá este commit ya que es un commit a modo de ejemplo.

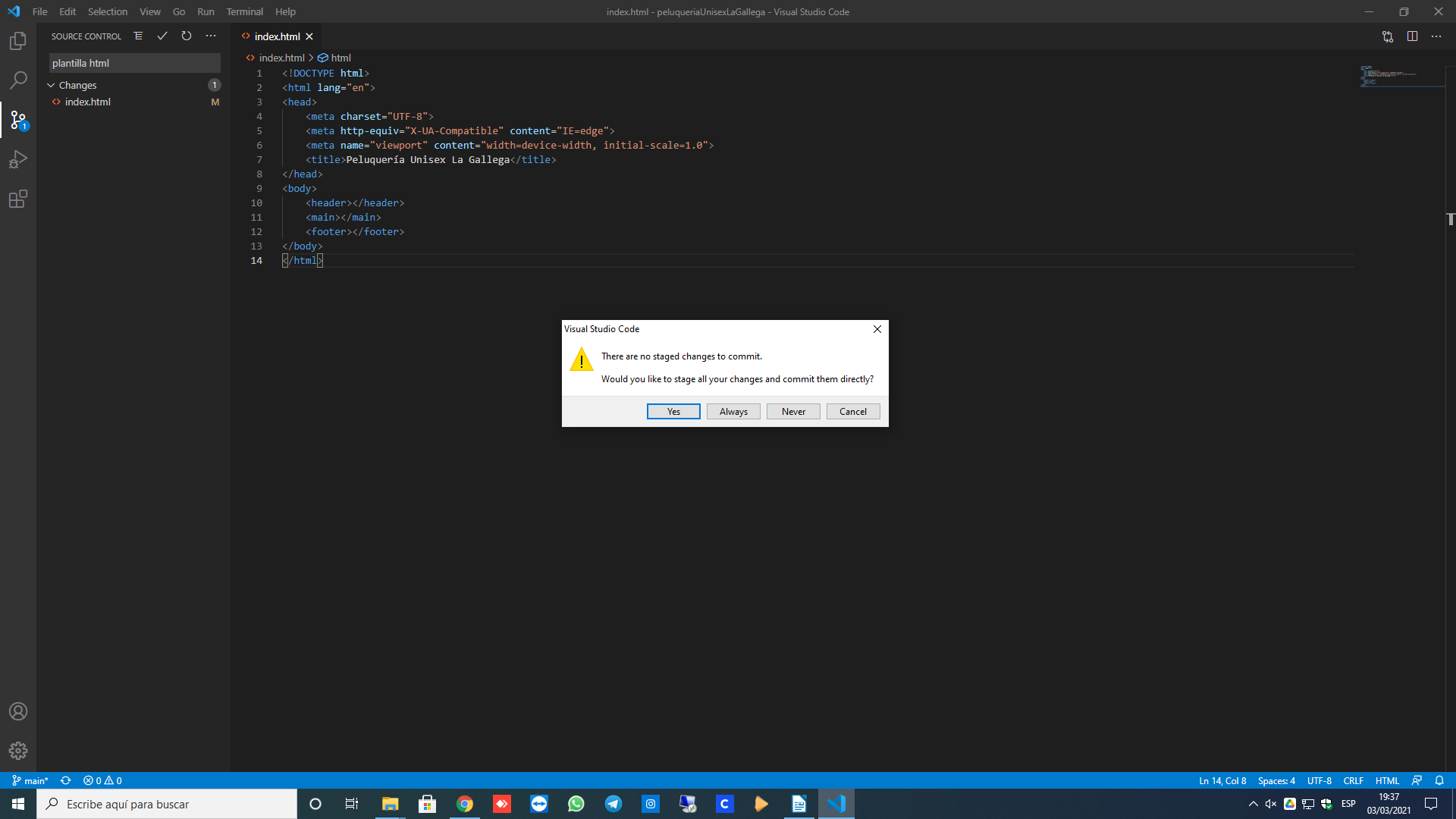
* En el menu de la izquierda hacemos click en el simbolo de la rama.



* Aquí nos indica los archivos que han cambiado, además, encima de eso hay una barra para escribir el mensaje del commit, lo escribimos y le damos al check.



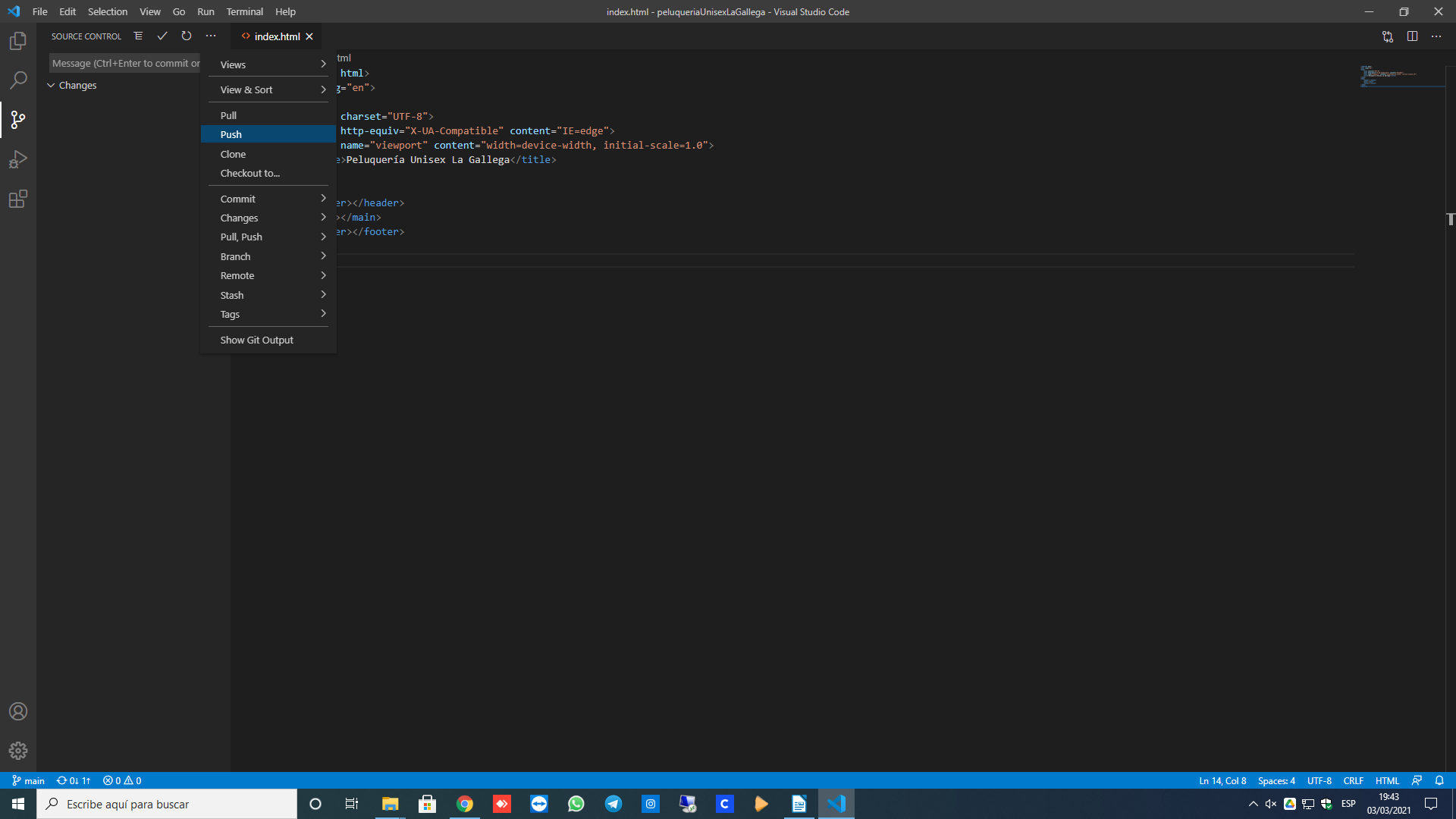
* Nos saldrá el siguiente mensaje al cual le diremos “YES”.



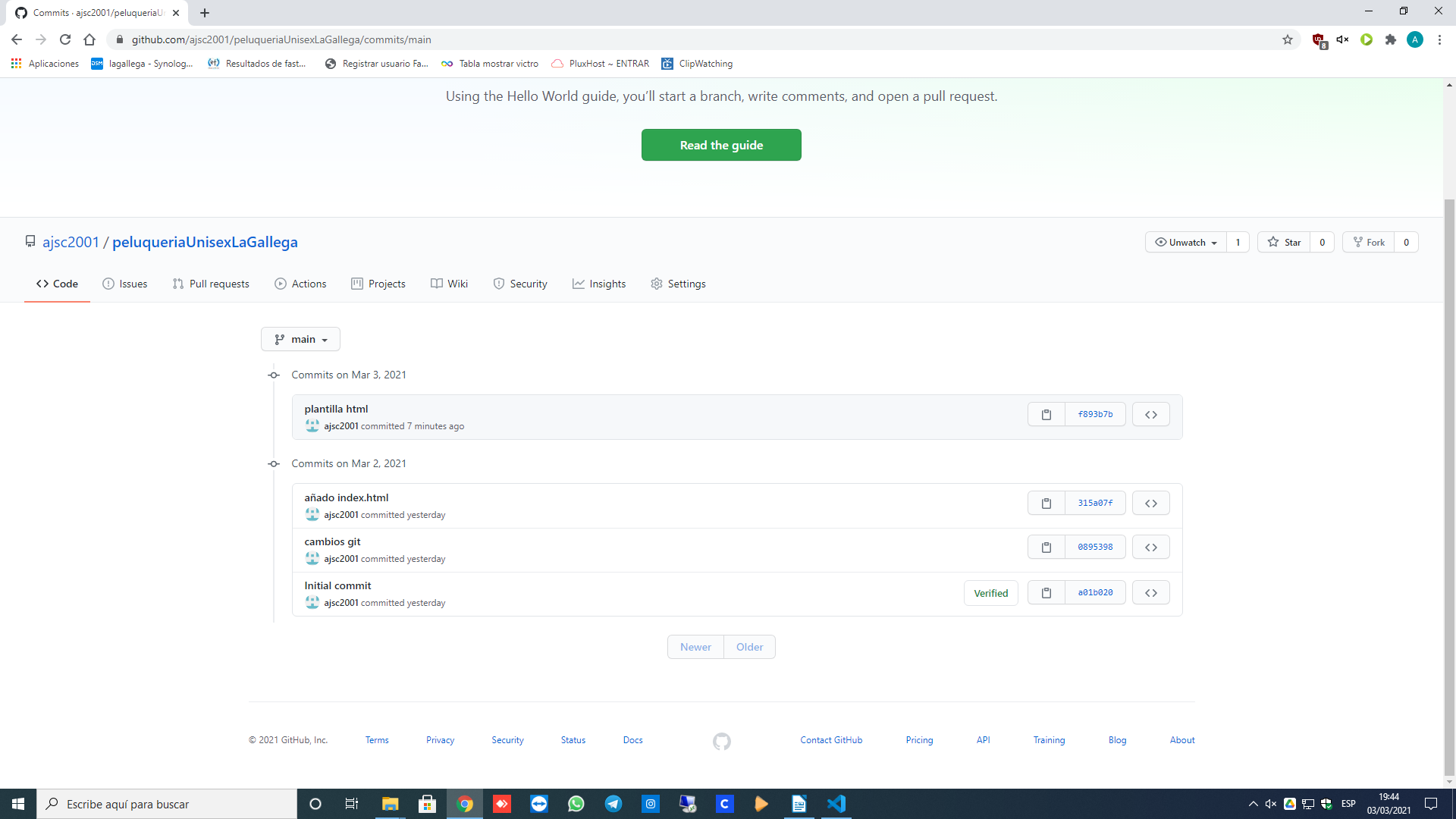
De este modo ya habríamos hecho el commit, pero como tenemos un repositorio remoto tenemos que actualizarlo, para ello tambien voy a usar VisualStudio Code en vez del comando: “git push”.

Para ello, realizaré los siguientes pasos:

* Desde la ventana en la que habia hecho el commit, pulso los 3 puntos que hay cerca de donde estaba el check.
* Selecciono la opción de push.



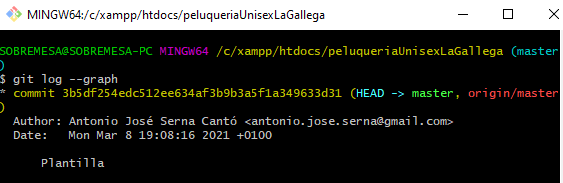
* Espero que acabe de cargar la barra de proceso y me dirijo a GitHub a comprobar si se ha subido.

Efectivamente, el commit que habiamos creado yua esta en GitHub.

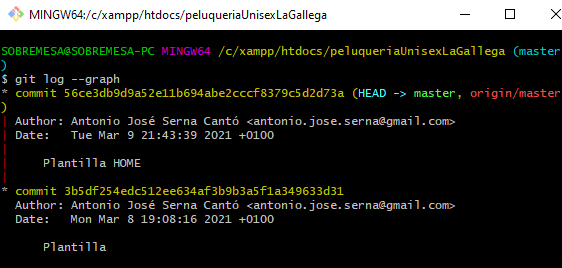
Este será el proceso que usaré para realizar todos los commit por lo que a continuación, cuando realice un commit solamente introduciré detalles sobre el commit realizado junto a la captura de la evolución.

## Registro de commits

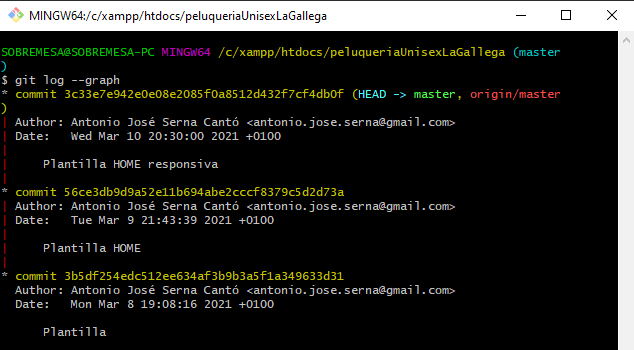
1. Plantilla: Empiezo a crear la estructura de la web



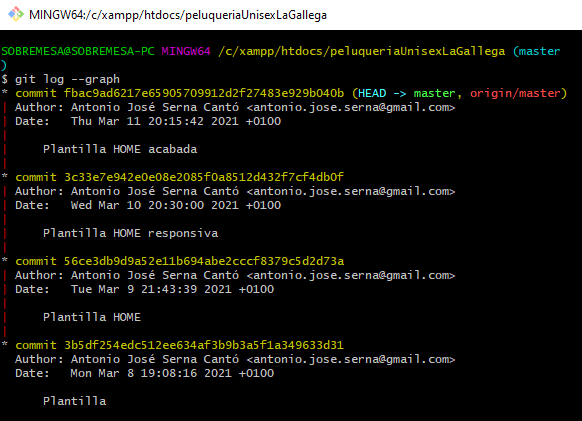
1. Plantilla HOME: Creo una platilla básica para la pantalla home usando imagenes de prueba y con el mínimo de elementos para asegurarme que funciona. Flata hacerla responsiva.



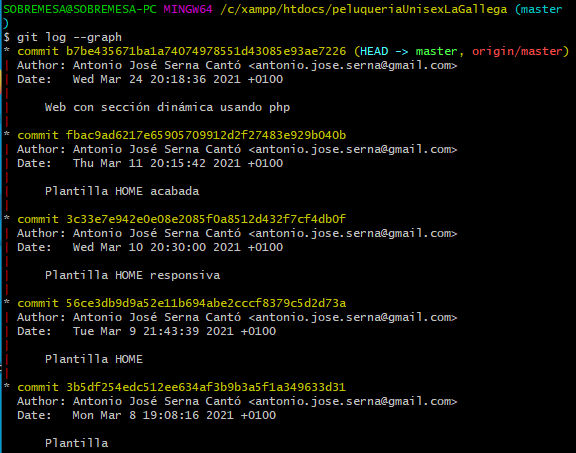
1. Plantilla HOME responsiva: Como indica este nombre, los cambios que hay en este commit con el anterior es que es responsiva.



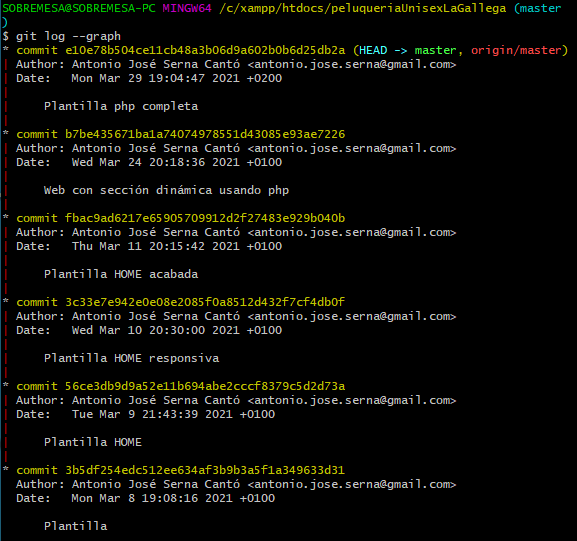
1. Plantilla HOME acabada: Como indica este nombre es una version de la ventana home, pero esta plantilla ya esta totalmente acabada y tambien es totalmente responsiva, solo faltaría meterle el contenido real de la web.



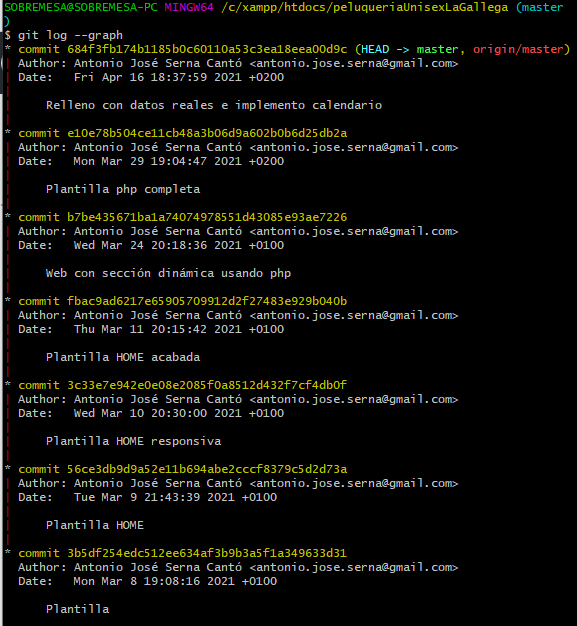
1. Web con sección dinámica usando php: En este commit ya tengo una web de ejemplo con todas las páginas creadas excepto las relacionadas con usuarios, por ejemplo: login, logout, etc.



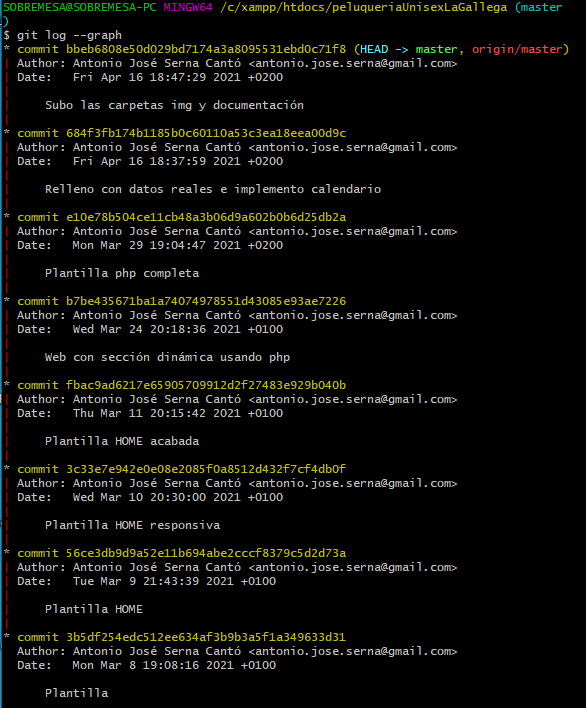
1. Plantilla php completa: En este commit se encuentra un plantilla funcional con todas las paginas que se espera que tenga la web.



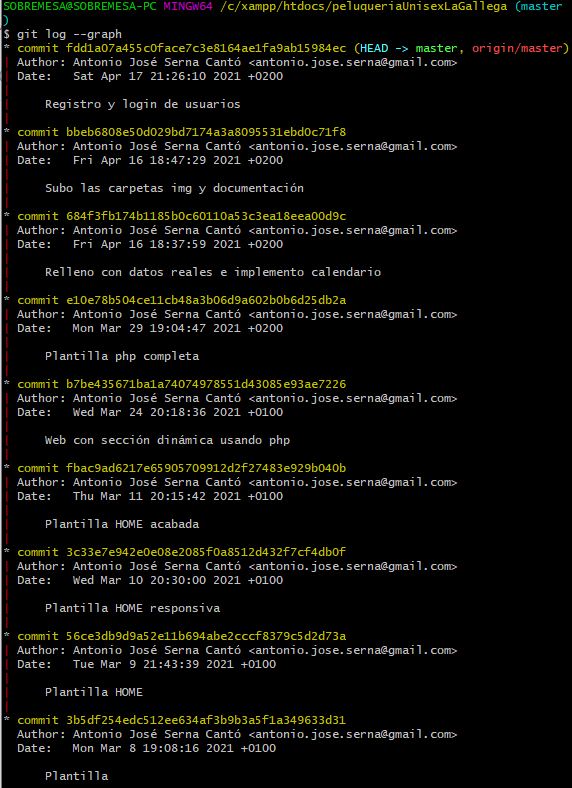
1. Relleno con datos reales e implemento calendario: En este commit lo que he hecho ha sido empezar a rellenar la web con datos reales e implementar un calendario con la librería Jquery UI. Además de esto he serguido haciendo ajustes de CSS.



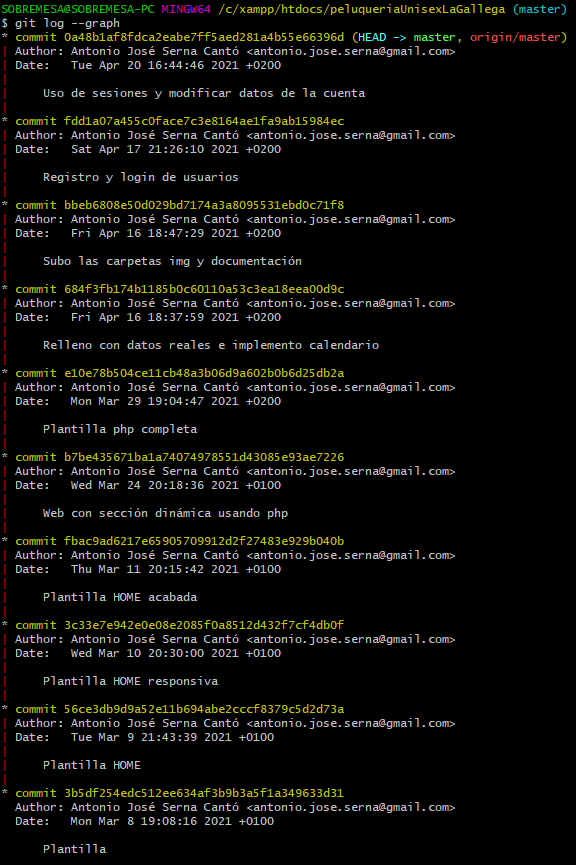
1. Subo las carpetas img y la documentación: En este commit quito del archivo .gitignore las carpetas *img* y *documentacion* para que se suban al repositorio.



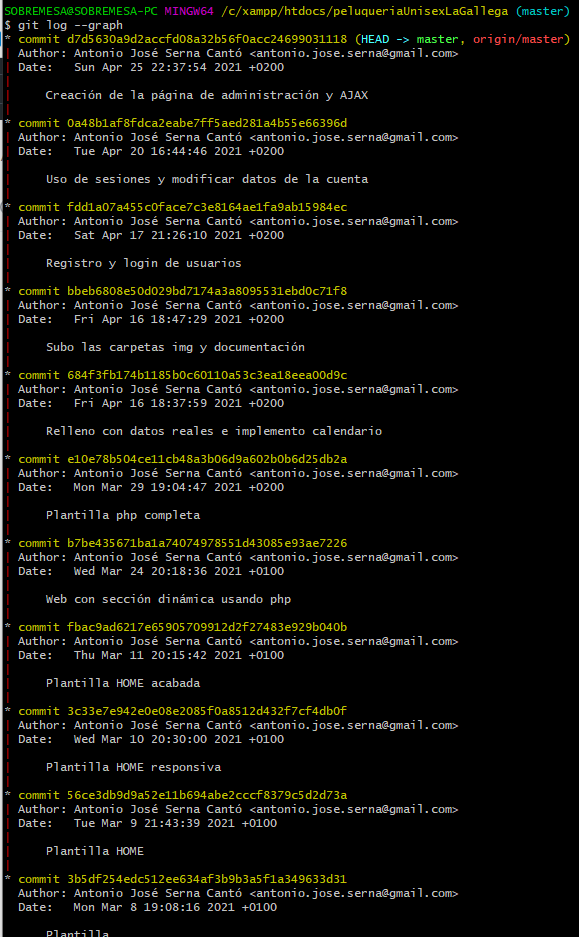
1. Registro y login de usuarios: Creo la clase usuario, añado funcionalidades como la codificación de contraseña y redacto las funciones para conseguir añadir un usuario a la BD y conectarme a la web con un usuario determinado.



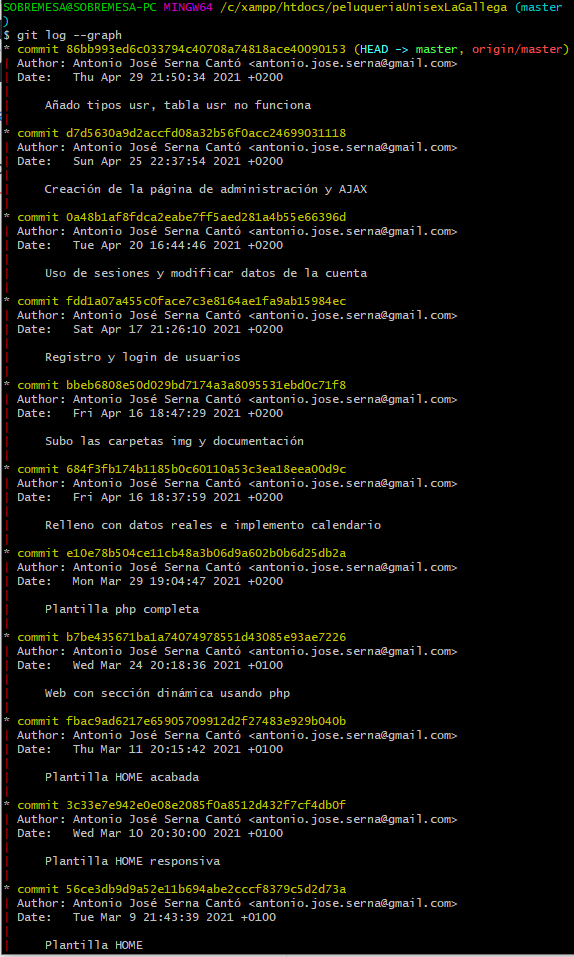
1. Uso de sesiones y modificar datos de la cuenta: En este commit he completado el commit anterior de inicaio de sesión con unas variables SESSION para almacenar el id, tipo de usuario y el nombre para posterior uso. También he implementado la funcionalidad de modificar datos del usuario y sus respectivas funciones y cambios necesarios como por ejemplo algunos valores de las variables SESSION.



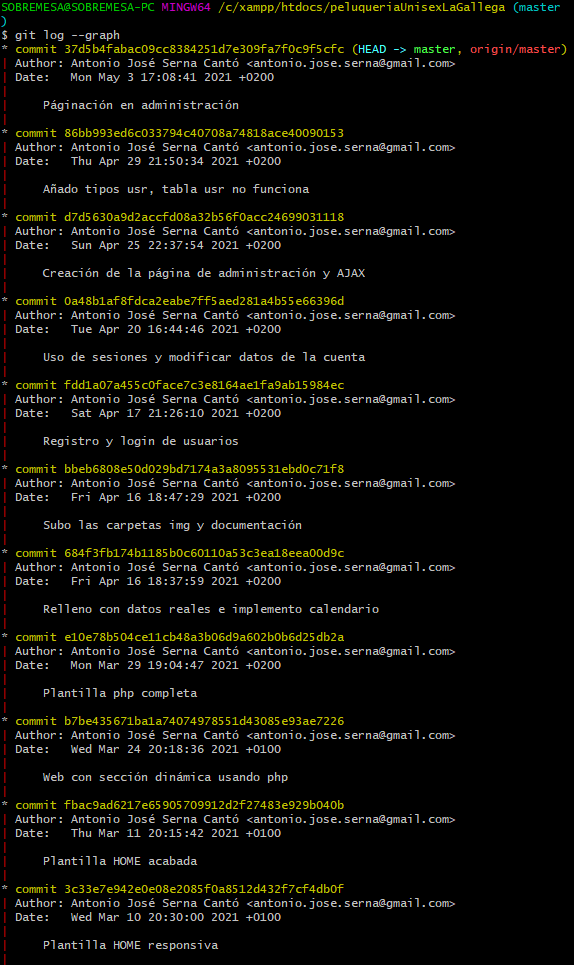
1. Creación de la página de administración y AJAX: Enn este comit se ha creado una página para administrar el sitio donde de momento se encuentran disponibles las opciones de establecer un horario o modificarlo y la opcion de añadir servicios. Para ellos se han creado las clases horario y servicio. Tambien se han hecho algunas llamadas a PHP desde AJAX como por ejemplo eliminar un usuario (con el que estas logueado). En este commit también he comenzado a mirar como obtener la fecha seleccionada del datapicker de Jquery UI y para ello he usado un objeto Date de JavaScript, esto aun no funciona del todo, estoy mirando de hacer una conexión AJAX para actualizar los valores que se muestran del calendario según el dia de la semana seleccionado.



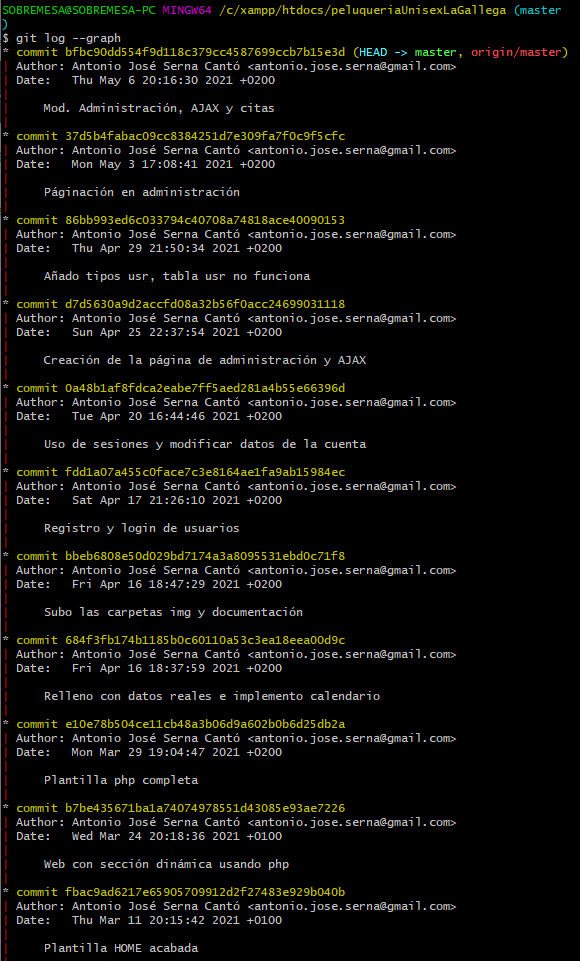
1. Añado tipos usr, tabla usr no funciona: Aplico la funcionalidad de tipos de usuario que ya lo tenia preparado, pero no activo, añado la tabla de usuarios en administración sin ninguna funcionalidad, solo mostrar datos. He decidido que el administrador solo podrá eliminar el usuario y modificar el tipo de usuario que es el usuario que va a modificar. En el caso de que el usuario pierda la contraseña, el usuario deberá solicitar al administrador una recuperación de clave, este, teniendo el correo del usuario ejecutará una funcion que esta fuera de la web para obtener todos los datos del usuario y de ese modo saber cuales son tanto su nick como su contraseña para indicarselo al cliente, despues queda bajo la responsabilidad del ciente cambiar dicha contraseña.



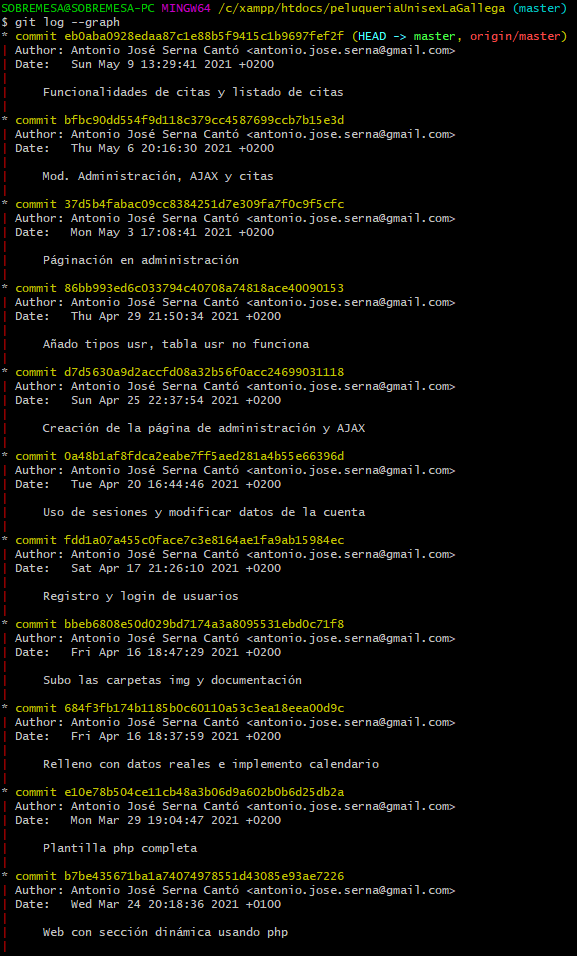
1. Páginación en administración: Se resuelven algunos errores de la página de administración y se implementa paginación en algunas opciones de esta. También se le añade un efecto con Jquery. Además de lo mencionado anteriormente también se han estado realizando pruebas sobre como obtener el horario de un dia mediante datepicker y conexión mediante AJAX.



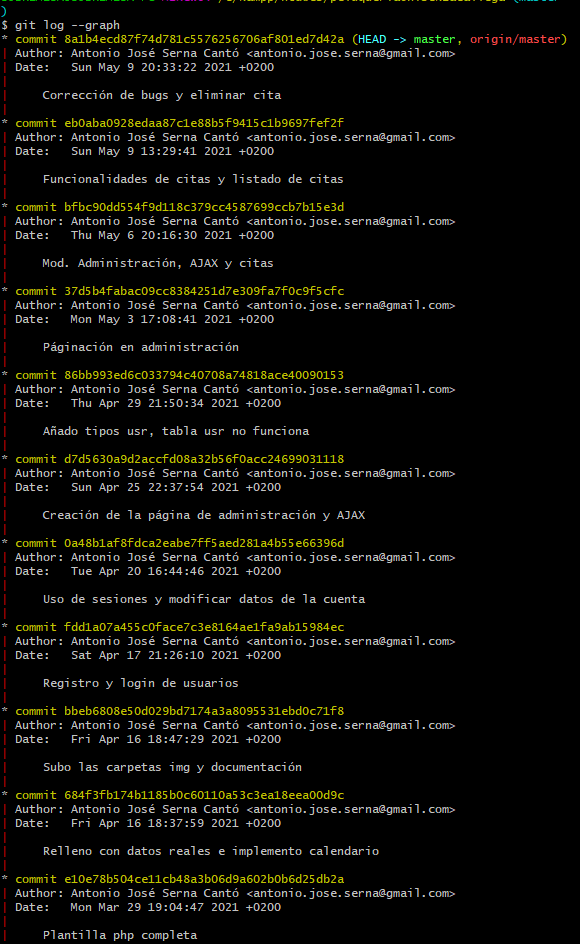
1. Mod. Administración, AJAX y citas: Commit numero 12 invalido el tema de contraseña por que ahora desde administración lo que se hace es dejar siempre ese campo vacio y si alguien solicita dicho cambio de contraseña entonces se rellena y de este modo cambiamos la contraseña. En este commit, como su nombre indica, se ha realizado cosas con AJAX, estas cosas ha sido solucionar los errores que daba al clickar en las papeleras de administración. También se han solucionado errores de paginación usando variables de sesiones tanto en JavaScript como en PHP y haciendo comunicación entre ellas via AJAX. Se ha añadido una funcionalidad que en la pagina donde se haga una sesion que solo tiene que ser funcional ahi, cuando clicke en algun sitio que no tenga que ver con eso se elimina tanto la sesion de JavaScript como la de PHP. Estos son algunos cambios entre otros cuantos que se han realizado para resolver algunos errores. Además he comenzado a realizar la parte de gestión de citas.



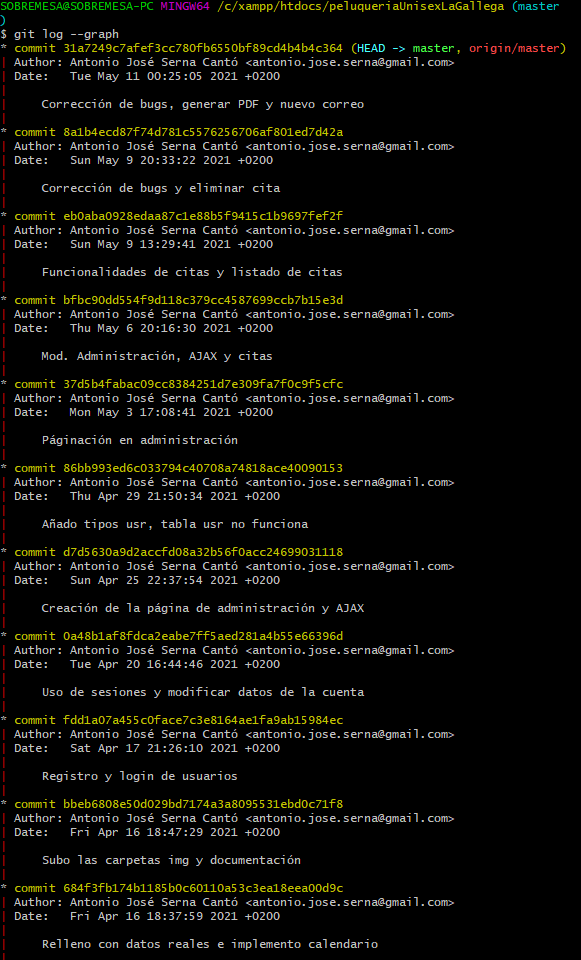
1. Funcionalidades de citas y listado de citas: Creación de las funcionalidades de citas y listado de citas. Se crea la clase cita para las conexiones con la BD. Se crean las funcionalidades de la página citas para poder reservar citas según las reservas existentes para dicho dia. Para saber cuales son las horas disponibles se ha tenido en cuenta la hora de inicio para la nueva cita, si daba teimpo a realizar la nueva cita mirando el inicio de la proxima cita o el fin de horario de mañana o tarde. En la página de reservas se muestra el listado de citas paginado según si eres ciente o no, si eres cliente obtienes el listado de todas tus citas, si no eres cliente entonces obtienes el listado de todas las citas existentes.



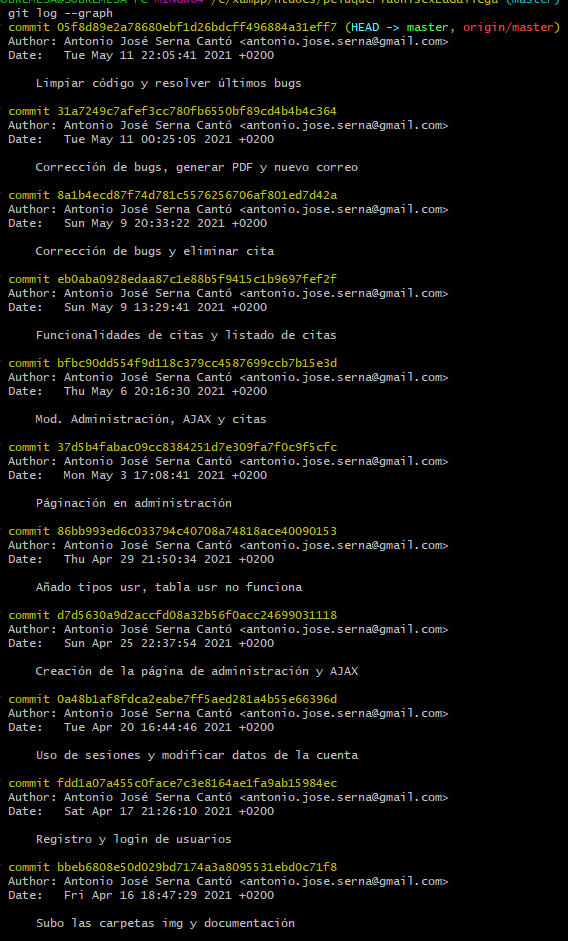
1. Corrección de bugs y eliminar cita: En este commit se han corregido algunos bugs que habian esparcidos por la página y se ha añadido la funcionalidad de eliminar una reserva.



1. Corrección de bugs, generar PDF y nuevo correo: Se han corregido algunos bugs como que te dejaba reservar cita para una fecha que yanabia pasado, te dejaba acceder a paginas de administración si ponis los parametros de url, etc. Además se ha añadido la librería ‘fpdf’ que es con la que genero el listado en pdf con la información de las citas. Tambien ha añadido el envio de un correo de confirmación cuando reservas la cita.



1. Limpiar código y resolver últimos bugs: Limpio código y añado comentarios necesarios para facilitar la legibilidad de este y se resuelven algunos pequeños bugs que existían como la paginación de las reservas.



1. Compatibilidad PHP hosting: En este commit simplemente se modifica un pequeño bug de lo que muestra las reservas a usuarios clientes y también se cambia la función array\_key\_last predefinida en PHP por una creada por mi con la misma finalidad ya que en el hosting no funciona dicha función de PHP.

