Un cliente encargado de realizar formaciones a empresas ha decidido crear un curso nuevo para poder enseñar aplicaciones javascript a interacciones con elementos HTML. Por eso, es necesario que implementemos una página web sencilla en la que mostremos algunos ejemplos en los que se puede utilizar este lenguaje de programación. La segunda idea es que el producto final tenga un aspecto similar al siguiente esquema:

(esto es una tabla):

Cabecera de la página con el número del curso y una imagen de fondo{

-Home

-Modificar HTML continuo

-Modificar Estilos CSS

-Modificar atributos HTML

}

Nuestro equipo de programadores se compone de 3 personas. Por este motivo, he decidido dividir el trabajo de la siguiente manera:

El usuario 1, que es el más experimentado, se encargará de crear el repositorio del proyecto e implementar la estructura inicial del proyecto utilizando el siguiente texto estándar

(enlace: https://github.com/h5bp/html5 boilerplate) . Esta página inicial Incluye la cabecera, la barra de navegación en el acceso a Home y la pequeña página.

En la parte central de la página inicial se incluye una breve descripción del curso

El usuario 2 creará la sección "Modificar contenido HTML" en la cual incluirá algún ejemplo similar al que existe en el siguiente enlace:<https://www.w3schools.com/js/tryit.asp?filename=tryjs_intro_inner_html> Así como una explicación del código.

También desarrollará la sección "Modificar atributos HTML".

El usuario 3, que acaba de llegar a la empresa, implementará la sección "Modificar estilos CSS" de la misma forma que ha hecho el usuario 2. Añadirá ejemplos similares a estos:

<https://www.w3schools.com/js/tryit.asp?filename=tryjs_intro_style>

(comentar el código).

Así pues, nuestra tarea consistirá en crear un repositorio en Github en el cual simultáneamente toda la implementación acorde a más de documentar todo el proceso correcto porque el nuestro cliente vea que son una empresa seria y trabajamos siguiendo una metodología.

Esta documentación, debera estar incluida en las páginas de github del propio repositorio del proyecto y tiene de contenido:

-Una introducción teórica de qué es git y la metodología utilizada. Por supuesto, la metodología utilizada no puede ser un "tot a master". Al Menos se debe utilizar una rama para producción y otra para

desarrollo.

-Explicación detallada del proceso realizado (capturas incluidas) en el cual:

-El usuario 1 crea el repositorio con los primeros ficheros acordados (estructura inicial) y adecua éste para poder utilizar la metodología elegida (creación de ramas iniciales).

-El usuario 2, crea dos características (feature/contenidoHTML y feature/atributosHTML, 1 por cada sección que se ha implementado.

El usuario 3, crea una característica (feature/estilosCSS) asociada al cambio para añadir la sección. (0,5 despejes)

El usuario 1, que es el más experimentado, se encargará de realizar las pruebas. Por este motivo, en el momento en que estén todos los cambios fusionados en la rama principal del proyecto, etiquetará la versión (v1.0) y creará una rama (test) que será la que supuestamente utilizará nuestros testers para probar el desarrollo.

Además, el usuario 1 ha decidido "automatizar" algunos procesos rutinarios. Por este motivo, decide crear los siguientes hooks de git:

-Automatización de la instalación del proyecto. Cuando se realiza un clonatge/checkout del proyecto, se ha de recrear la carpeta de node\_modules del proyecto.

-Verificación del formato mensaje de commit. Siempre que se realice un commit, el mensaje asociado ha de contener:

\*Un campo MOTIVO DEL COMMIT: Título asociado al commit.

\*Un campo IMPLEMENTACIÓN: con la explicación del desarrollo a realizar.

-Comprobación de caracteres extraños. Verificará que ninguno de los archivos que se vaya a subir contenga texto con caracteres extraños como àáâäèéêëìíîïòóôöùúûüñç

(y las mayúsculas).

-Antes de realizar el commit, se verificará el formato correcto en los instaladores html.

Además, se ejecutará el linter eslint utilizando el siguiente plugin: <https://www.npmjs.com/package/eslint-plugin-html>.

-Debido a que los hooks anteriores están disponibles para el resto de miembros del equipo,

el usuario 1 los subirá en una carpeta del proyecto denominada gitHooks

-Para finalizar el trabajo nuestro repositorio tendrá que contener, además de las ramas necesarias para poder realizar el desarrollo comentado, una rama (gh-pages) en la cual se incluirá toda la documentación del proceso y que será la que publicar en las páginas de github