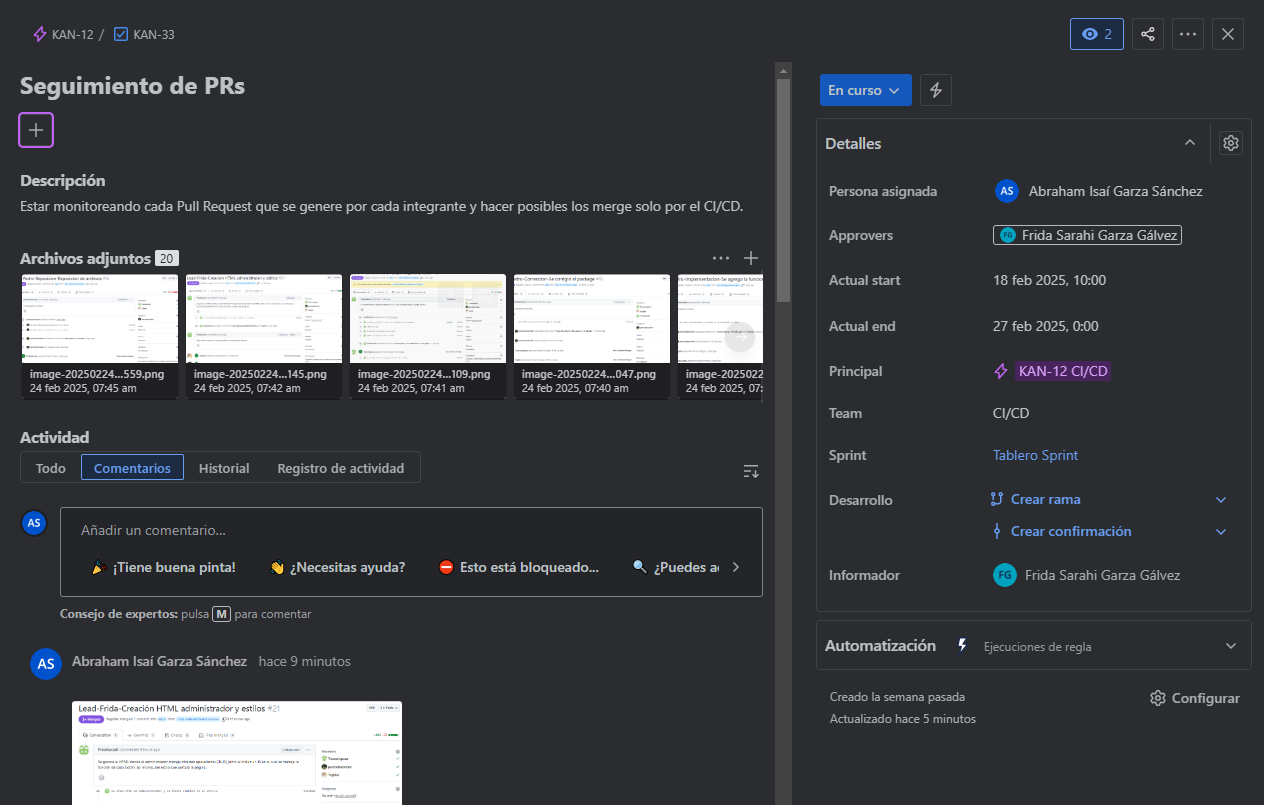
## Abraham Isaí Garza Sánchez

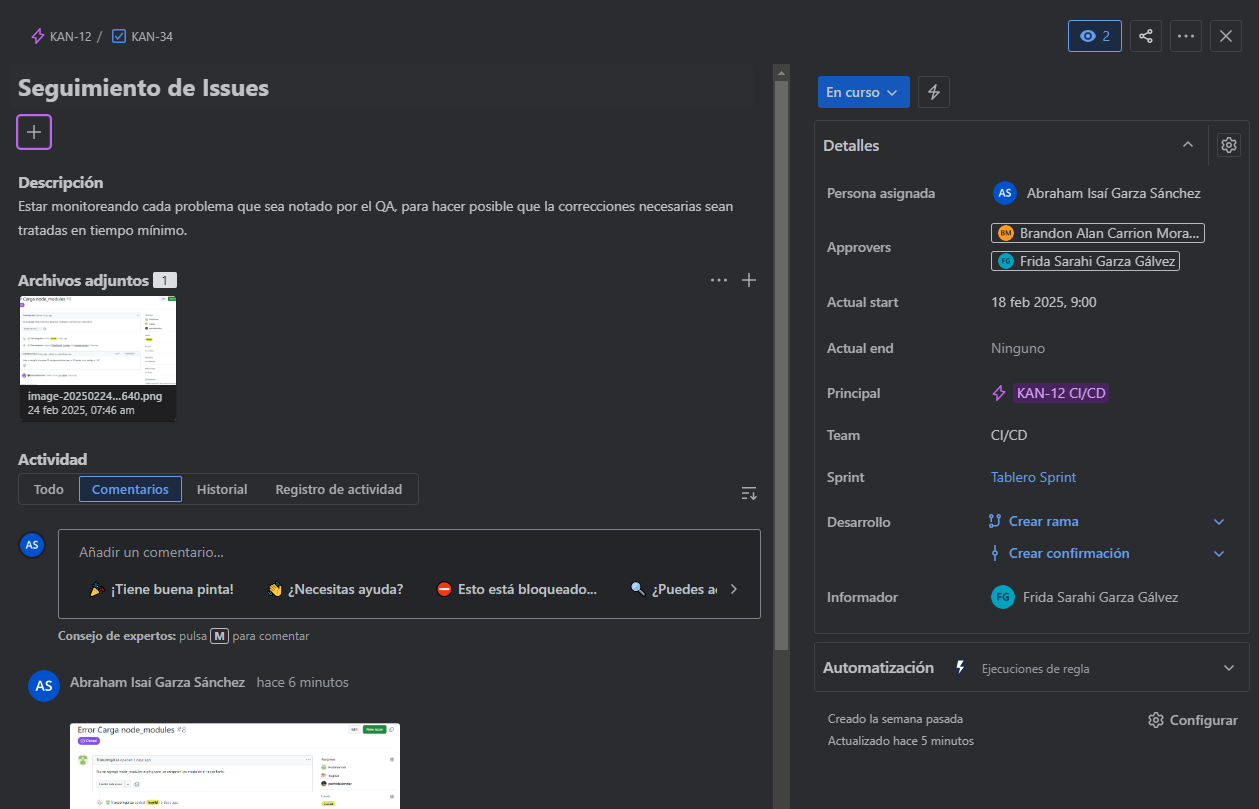
## - CI/CD

*El rol que se me asigno se encargo principalmente de llevar a cabo el manejo del Repositorio del Proyecto Final, también se encargó del despliegue de la pagina por medio de la nube*

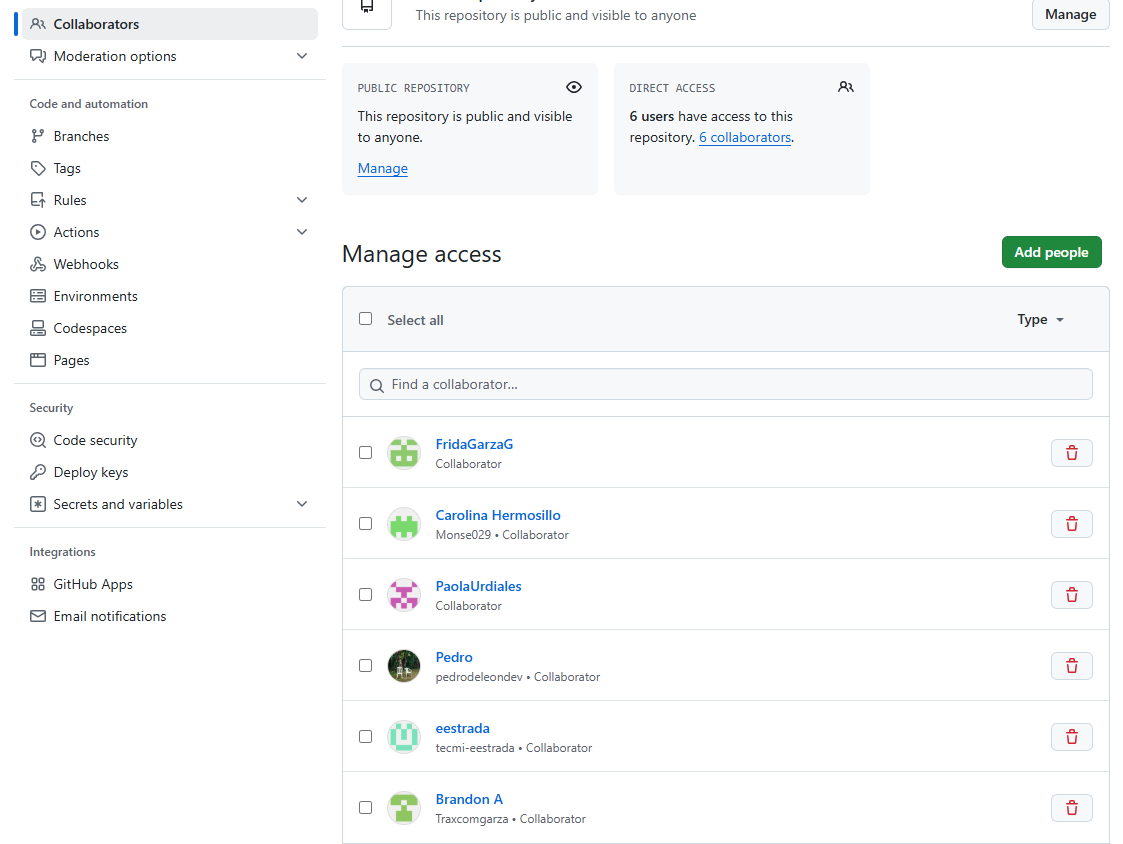
#### Requerimientos

* Se debía llevar a cabo el seguimiento de Pull Requests dentro del repositorio, estos iban siendo documentados a través de imágenes por medio de Jira.



* Se debía llevar a cabo el seguimiento de Issues dentro del Repositorio, estos también fueron documentados a través de imágenes por medio de Jira.
* Se debía hacer el despliegue de la aplicación a través de una plataforma como Heroku, AWS, Digital Ocean o Render.

#### Realización

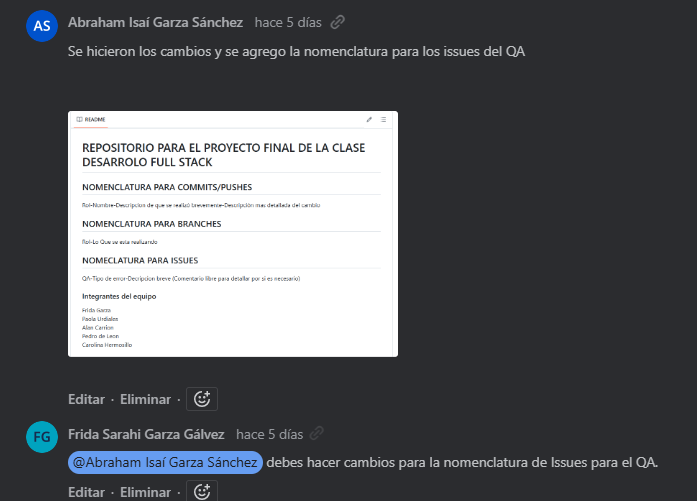
Lo primero dentro de la realización del proyecto para el CI/CD fue la creación del Repositorio que fue nombrado “Proyecto-Final”. El paso a seguir fue incluir a los miembros participantes del equipo dentro del repositorio, así como añadir al profesor para que tenga acceso a este y pueda verificar/corroborar los resultados.

Después se realizó la creación del archivo README.txt dentro del repositorio para facilitar la presentación de esté. En esta primera iteración del archivo se incluyeron la nomenclatura que debía ser usada para los Commits y la nomenclatura que debía ser usada para los Branches que se creen dentro del repositorio. También se escribieron los nombres de los integrantes del equipo.

En este paso se tuvo el primer Problema que será abordado dentro de sección de problemas dentro de este mismo documento.

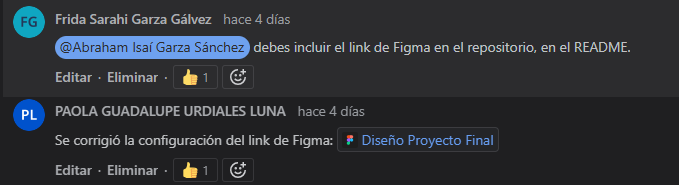
Después de esto se creo una carpeta dentro del repositorio donde se colocarían los documentos necesarios para que los miembros del equipo puedan subir sus respectivos reportes o documentos sin la necesidad de que entren interfieran con la raíz del repositorio. Esta después fue renombrada por el LEAD a “ProyectoFinal-Documentos” donde el LEAD subió un archivo base de como se debían realizar los reportes del equipo.

Después de esto, se me fue pedido a través de Jira por el LEAD incluir la nomenclatura para los Issues dentro del repositorio ya que este no había sido escrito anteriormente.

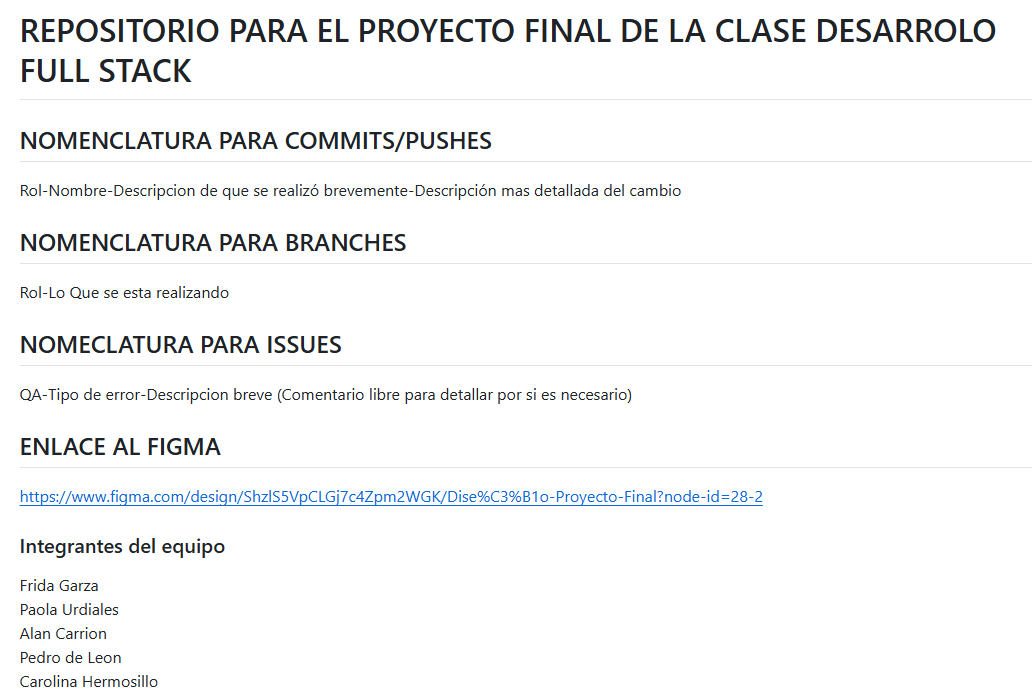


Este fue implementado correctamente y sin problemas, se opto por usar “QA-Tipo de error-Descripción breve (Comentario libre para detallar por si es necesario)” como nomenclatura para los Issues.

Después se me fue solicitado por el LEAD incluir el Link del Figma dentro del README del repositorio para facilitar su acceso.

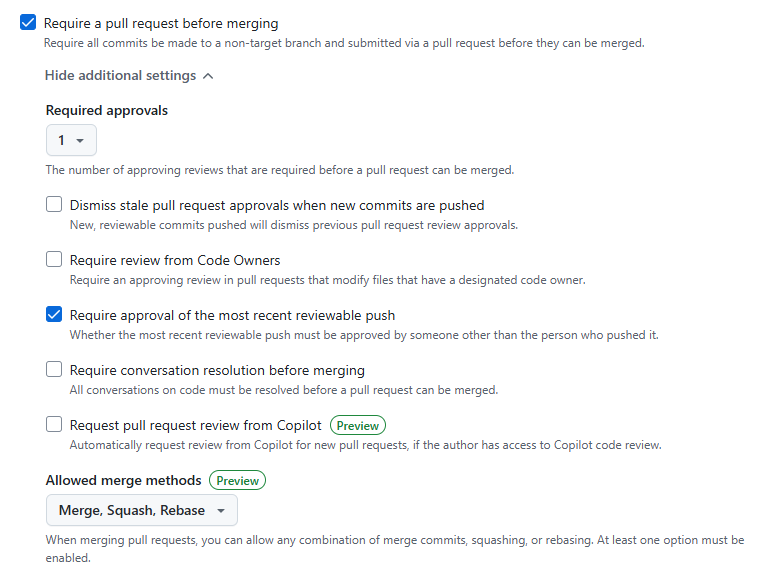


Finalmente, el README del repositorio se vio de la siguiente manera.

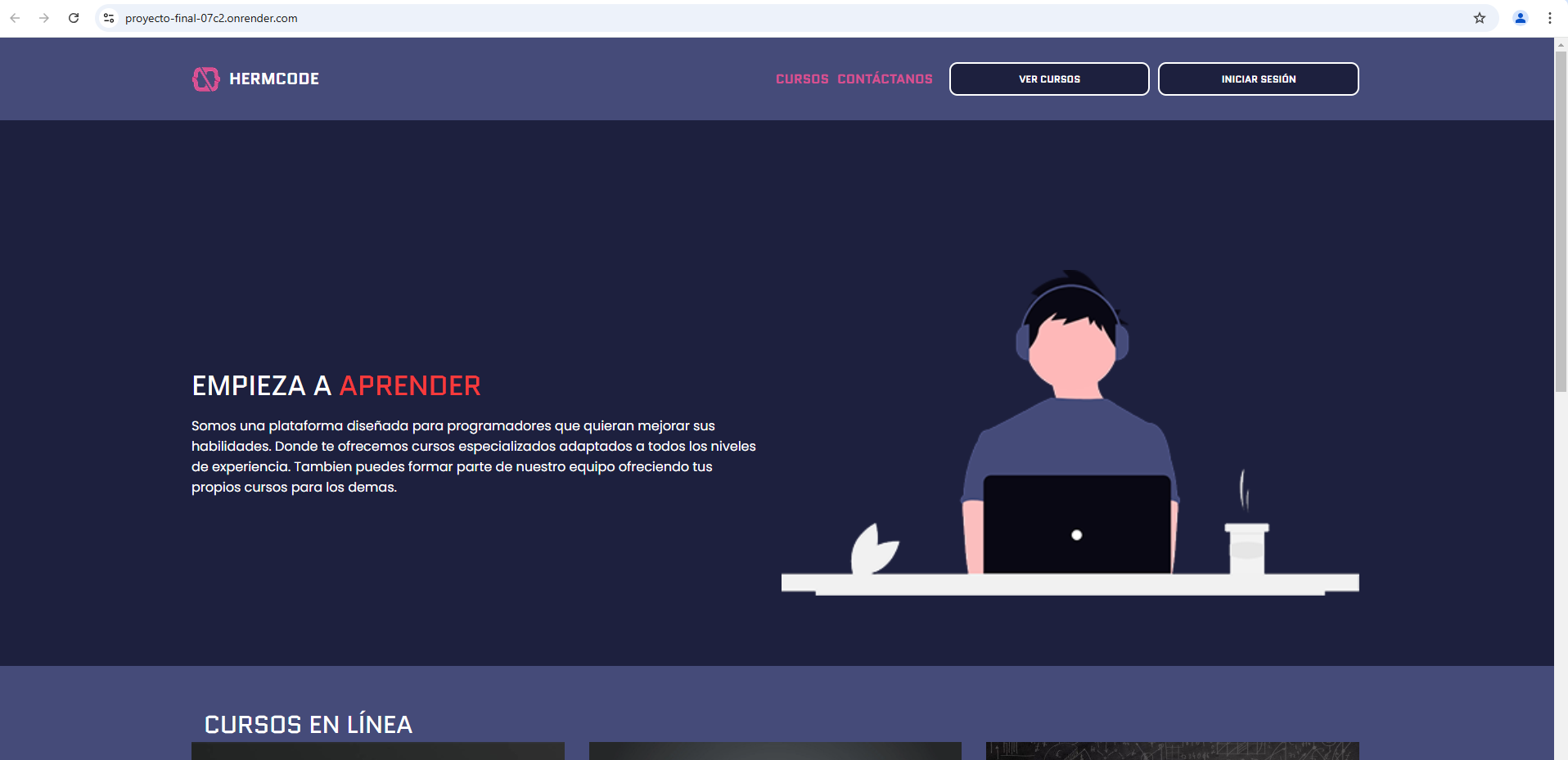


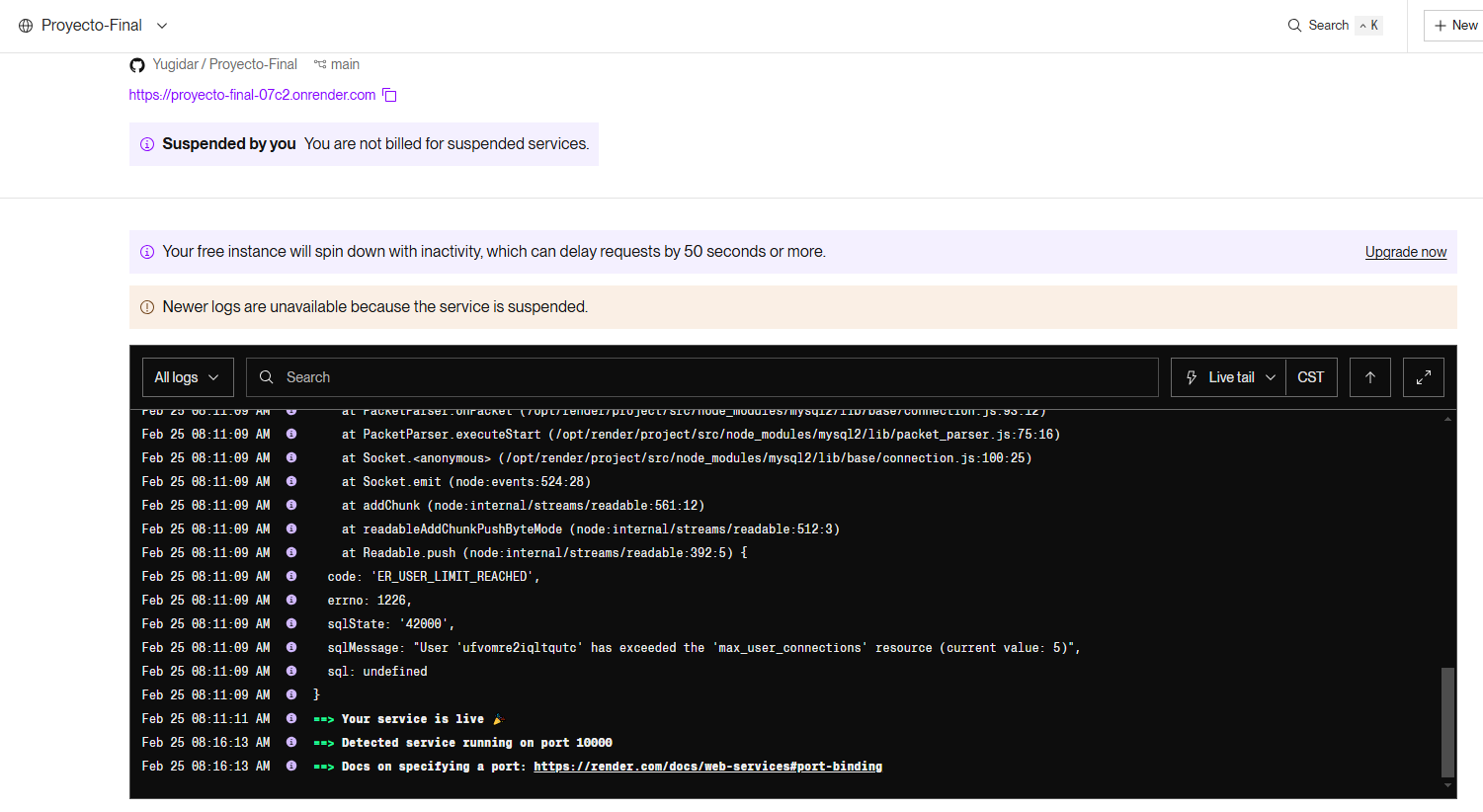
Añadido a esto dentro del repositorio se modificaron las reglas de este para que al momento de querer fusionar los Branches a la rama principal este tenga un bloqueo y no pueda realizarse el Merge sin que antes tenga por lo menos un Approve de algún reviewer asignado, esto para evitar problemas al momento de realizar Merge por roles que no sean CI/CD.



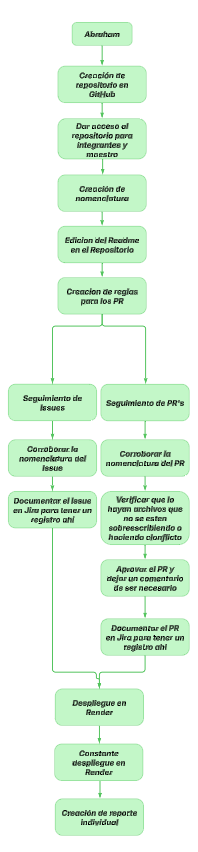


**Despliegue en la nube**  
Para el despliegue en la nube se considero utilizar diversos servicios como AWS, Digital Ocean, Heroku y Render. Al final se opto por utilizar Render debido a su interfaz de fácil uso y que permite desplegar la pagina directamente desde nuestro repositorio de GitHub sin la necesidad de registrar alguna tarjeta de crédito o débito o de algún tipo de pago. La página hace una configuración inicial basada en los archivos que encuentra en el repositorio para poder desplegarla, también cuenta con otros planes para alojar paginas mas pesadas, es decir que necesiten de bases de datos grandes o que tengan mucho tráfico. Una vez seleccionado el repositorio de nuestro proyecto (Proyecto-Final) hizo la configuración inicial basada en los archivos encontrados, que fue aquí donde se formo el segundo error que será abordado en su debida sección. Una vez arreglado el problema la pagina se desplego de manera correcta utilizando el siguiente enlace: https://proyecto-final-07c2.onrender.com/  
con la consola y pagina viéndose de la siguiente manera.





También se realizó la el diagrama de actividades para CI/CD dentro de Lucid, basándose en las actividades realizadas y ya documentadas en este documento para verse de la siguiente manera



#### Inconvenientes/Problemas y Solución

1. El primer problema fue el formato del archivo README.txt, el cual, aun que se le implementaron Headers para una visualización mas placentera, estos no se veían reflejados dentro de la página principal del documento. Después de buscar por la causa de esto, se pudo resolver al cambiar la extensión del archivo, es decir, se cambio de README.txt a README.md.
2. El segundo problema que fue que al momento de querer desplegar la pagina en la nube con la configuración inicial que te da render arrojaba el siguiente error: “Error: Cannot find module '/opt/render/project/src/index.js'”. La suposición inicial fue que no teníamos un archivo llamado index.js dentro del repositorio así que se consulto con el Dev si en realidad teníamos este archivo, después de corroborar que no contábamos con ningún archivo llamado index.js se siguió indagando dentro de los foros de Troubleshooting de Render; tras no dar con el problema se recurrió al uso de una inteligencia artificial para verificar si podía resolver el error. La primer sugerencia fue verificar que la estructura del proyecto fuera la correcta, con package.json fuera de la carpeta de source, tras verificar que la estructura era la correcta lo siguiente a corroborar eran los comandos con los que se estaba iniciando la página. El error radicaba ahí, donde el comando de inicio (Start Command) estaba con un valor predeterminado de algo así como “/src/index.js”, este tuvo que ser cambiado a npm start para que la pagina pueda ser desplegada de manera correcta en la nube.
3. El tercer problema se presento al momento de querer utilizar una API que mostraba el país de donde estabas visitando la página, el problema era que al mostrar el país del que se conectaba el usuario mostraba un valor default. Después de buscar la causa del problema se concluyo que esto se debía a la manera en la que el API forzaba una conexión HTTPS mientras que el despliegue de la pagina debía utilizar API que utilizaran HTTP, esto se resolvió cambiando a otra API que funcionalmente hacia lo mismo pero sin la necesidad de que utilizara HTTPS