

Develop your terminal

Yogesh Yadav, Estefania Cordeiro, Raúl Cátedra, José R. Serralvo

Agradecimientos

Gracias al equipo docente y al resto de grupos por su apoyo y asesoramiento durante el desarrollo del proyecto. Gracias también a los integrantes de este grupo, la colaboración y la comunicación han sido fundamentales para conseguir los objetivos y desarrollar este proyecto.

Introducción

En el presente proyecto se va a intentar desarrollar una terminal funcional. En cada uno de los pasos, habrá que desarrollar las distintas funcionalidades de los comandos, además de implementar una visual similar a una consola y una serie de atajos de teclado.

Objetivos

Los objetivos de este proyecto se dividen en objetivos generales y objetivos específicos. A continuación, se exponen los distintos objetivos generales junto con sus correspondientes objetivos específicos.

- Diseñar una terminal funcional.
 - Crear todo el contenido en un solo HTML de forma dinámica.
 - Evitar que la página sufra recargas.
 - Utilizar solo HTML, CSS y JavaScript básico, sin frameworks ni librerías.
 - Comprobar que el contenido se muestra de manera similar en las últimas versiones de Chrome, Safari, Firefox y Edge.
 - Generar el diseño en Mobile first.
- Crear las funcionalidades de los distintos comandos.
 - Crear la funcionalidad de *pwd*.
 - Crear la funcionalidad de *ls*.
 - Crear la funcionalidad de *ls -R*.
 - Crear la funcionalidad de *ls -S*.
 - Crear la funcionalidad de *ls -t*.
 - Crear la funcionalidad de *cd*.
 - Crear la funcionalidad de *cd ...*.
 - Crear la funcionalidad de *mkdir*.
 - Crear la funcionalidad de *echo*.
 - Crear la funcionalidad de *cat*.
 - Crear la funcionalidad de *rm*.
 - Crear la funcionalidad de *mv*.
 - Crear la funcionalidad de *clear*.
 - Crear la funcionalidad de *help*.
 - Crear la funcionalidad de *man*.
 - Crear la funcionalidad de *JS*.
 - Implementar de tres a cinco comandos de nuestra elección.
- Crear las funcionalidades de los distintos atajos de teclado.
 - Crear la funcionalidad de la tecla arriba.
 - Crear la funcionalidad de la tecla abajo.
 - Crear la funcionalidad de la tecla tabulador.
 - Implementar, al menos, un atajo de teclado de nuestra elección.

Metodología

A continuación, se expone el conjunto de procedimientos a seguir para la consecución de los objetivos antes definidos.

- **Canales de comunicación.**

Se definen y crean canales de comunicación para la realización del proyecto. Específicamente, los canales utilizados son:

- *Microsoft Teams*, para poder hacer llamadas entre los integrantes del grupo.
- *Github*, para realizar el control de versiones del código que crea cada uno de los integrantes.
- *Github kanban*, para realizar la asignación de tareas a los distintos integrantes del grupo y el control de las mismas.

- **Tareas.**

Las tareas se definen de forma conjunta entre todos los integrantes del equipo de trabajo, haciendo una división clara de las partes del proyecto para poder afrontarlo de forma ordenada.

La asignación de las tareas se hace a través del canal de comunicación *Github kanban*, en el que el grupo asigna las tareas de forma equitativa a los integrantes, de manera que todos hagan tareas que incluyan diseño y escritura de código.

La ejecución de la mayoría de tareas se lleva a cabo de forma local por los miembros del equipo, existiendo tareas específicas, en las que estos desarrollos locales se unen en un único proyecto.

- **Escritura de código.**

Para la escritura de código se decide utilizar *Visual Studio Code*, ya que todos los integrantes del grupo conocen bien sus funcionalidades.

Desarrollo del proyecto

Github kanban

En esta herramienta se generan las tareas previamente definidas (ver Anexo 1). Gracias a sus funcionalidades, las tareas se asignan a los miembros del equipo y se van intercambiando entre las distintas columnas del *kanban* según su momento de desarrollo.

Log diario

A continuación, se desarrolla el *log* diario, dividido por los días en los que se ha estado trabajando en el proyecto, incluyendo en cada día las tareas realizadas y quién las ha desarrollado.

- Viernes 4 de diciembre de 2020: la tarde del primer día se dedicó en su totalidad a la estructuración y definición del proyecto, generando en común una lista de tareas con sus especificaciones generales y particulares, trasladando todo este contenido a la columna *brainstorming* de *Github Kanban*. (ver Anexo 2). Tras esta primera reunión, se implementó el diseño básico de la terminal, además de un esquema de las funciones principales.
- Miércoles 9 de diciembre de 2020: al inicio de la segunda jornada, se realizó una reunión grupal en la que se asignaron las diferentes tareas a realizar ese día (ver Anexo 3). Una vez asignadas, se llevaron a cabo las siguientes tareas: creación de la funcionalidad de guardado de un historial de comando utilizados, así como los atajos de teclado que nos permiten acceder al historial, se inició la creación de las funcionalidades de los comandos *cd*, *cd ..*, *ls*, *ls -R*, *ls -S*, *ls -t* y la creación del objeto que simula el árbol de directorios.
- Jueves 10 de diciembre de 2020: al inicio de la tercera jornada, se realizó una reunión grupal en la que se asignaron las diferentes tareas a realizar ese día (ver Anexo 4). Una vez concluida, los integrantes comenzaron a trabajar en las distintas tareas. La mañana transcurrió sin más novedades. Al finalizar el día, las funcionalidades de los comandos *mv* y *mkdir* estaban creadas.
- Viernes 11 de diciembre de 2020: tras la habitual reunión en la que se asignaron tareas (ver Anexo 5), se completaron las funcionalidades de los comandos *rm*, *cat*, *echo*, *man* y *help*, además de la creación del *metadata*. Por otra parte, se implementó el comando *cmatrix*, primer comando de elección propia.
- Lunes 14 de diciembre de 2020: se crearon los comandos *ckirby*, *JS*, *rch*, *y*, *n* y *ls -RTD*. Se finalizó la creación de las funcionalidades de los comandos *ls*, *ls -R*, *ls -S*, *ls -t*, *cd* y *cd ...*. Tras esto, se implementó el atajo de teclado de la tecla tabulador y los atajos de teclado de elección propia. Una vez hecho todo esto, se procedió a una búsqueda y reparación de *bugs*, dando por finalizada la escritura de código (ver Anexo 6). Con la llegada de la tarde, se comenzó la creación de la documentación adjunta para el proyecto.
- Martes 15 de diciembre de 2020: se finalizó la documentación y se adjuntó al proyecto.

Resultados obtenidos

Los resultados obtenidos tras el desarrollo del proyecto fueron los esperados. Se cumplió con los objetivos mínimos, así como con todos los extras. Es por ello por lo que se puede afirmar que el equipo ha desarrollado una terminal basada en un solo archivo *HTML*, en el que el código se inyecta de forma dinámica. Esta página es *responsive*, está creada en *mobile first*, no sufre recargas durante su funcionamiento y es compatible con las últimas versiones de *Google Chrome*, *Mozilla Firefox*, *Microsoft Edge* y *Safari*.

De la misma forma, se puede afirmar que se han trazado los pasos necesarios para el correcto uso de todos los comandos, cumpliendo con los objetivos secundarios. Por último, se ha cumplido de igual manera con la creación de los atajos de teclado y sus objetivos secundarios.

Lecciones aprendidas

- Uso y práctica de *windows.addEventListener*.
- Utilización de objetos y clases.
- Utilización de consola y sus posibles tipos de rutas.
- Ajuste de contenido de una página solo con *flexbox*.
- Inyección de código *HTML* en *JavaScript* con *innerHTML*.
- Realización de cambios de rutas mostradas dependiendo de los comandos.
- Manipulación y cálculos con fechas.
- Utilización del *LocalStorage*.
- Manipulación de datos para adecuarlos a nuestras necesidades.
- Utilización de *return* para obtener un resultado que mostrar en pantalla.
- Utilización de *new Error* para mostrar errores.
- Utilización de *try and catch*.
- Utilización de *eval*.
- Utilización de *Github*.

Problemas encontrados

- Simular directorios a través de un objeto nos ha generado la necesidad de una gran manipulación de los datos para poder usar todos los tipos de rutas posibles.
- Utilizar el evento *keydown* para habilitar el autocompletado con la tecla tabulador.
- Crear un objeto con el *metadata* de las rutas para poder ordenarlas por tamaño y momento de creación
- Utilización de *Github*, por ser el primer proyecto creado en su totalidad en esta plataforma.

Problemas internos del equipo




























El desarrollo del proyecto ha sido fluido y cordial, por lo que no han aparecido problemas de ningún tipo entre los integrantes del equipo, ya sean de comunicación o de diferencias de opinión no resueltas.

Conclusiones

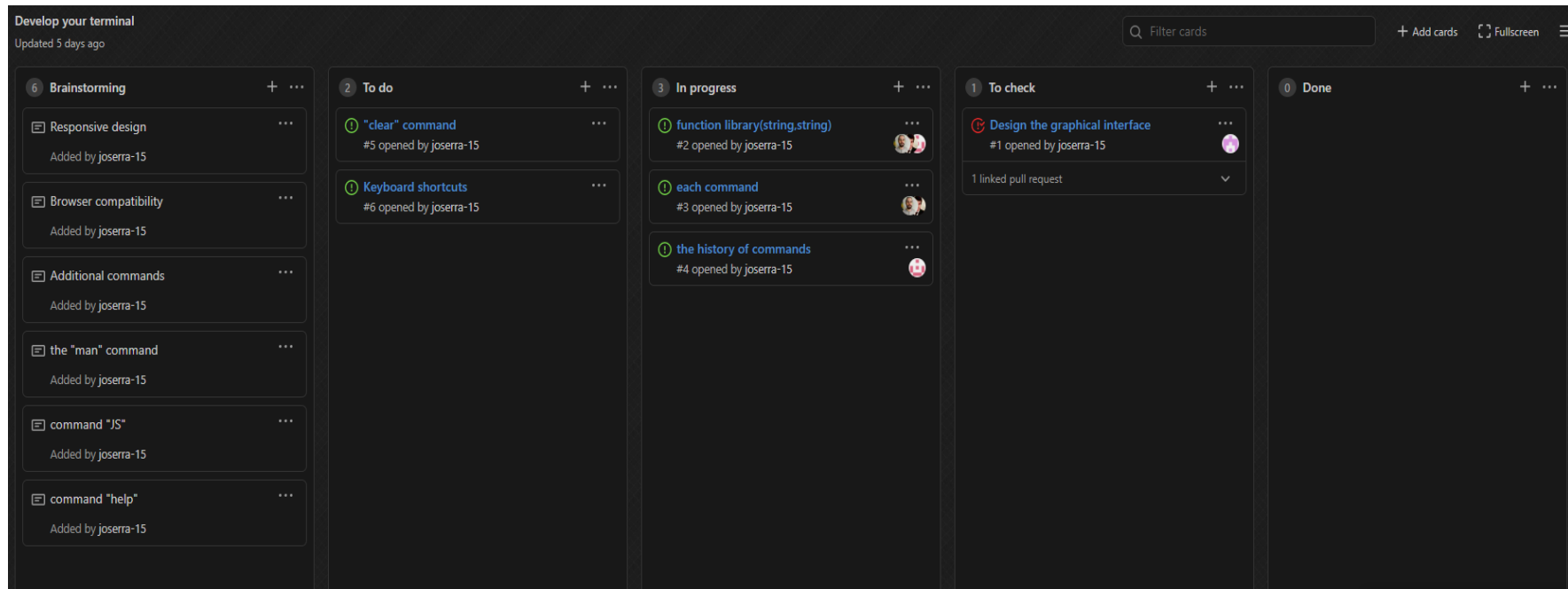
En conclusión, hemos realizado un proyecto bastante completo, cumpliendo con los objetivos marcados desde el principio. Cabe destacar que el proyecto, debido a las elecciones tomadas, no es muy escalable a proyectos mayores.

Anexos

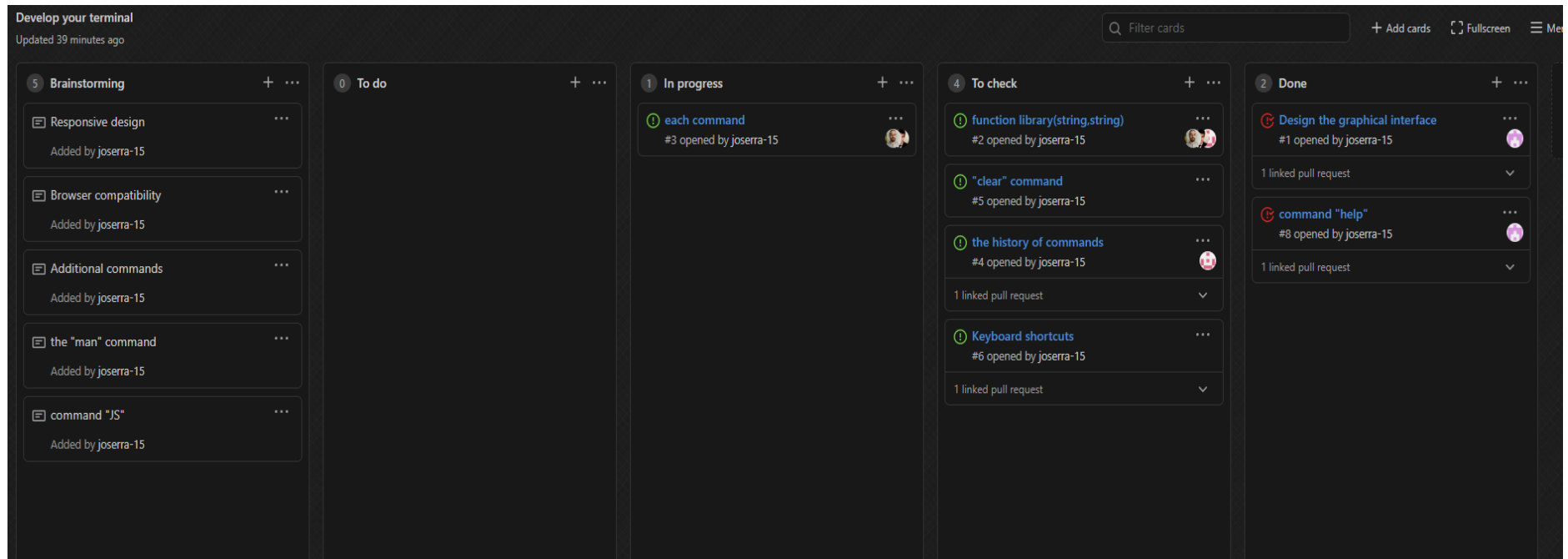
Anexo 1. Definición de tareas

 Design the graphical interface ... #1 opened by joserra-15  1 linked pull request 	 command rm ... #12 opened by joserra-15  1 linked pull request 
 command "help" ... #8 opened by joserra-15  1 linked pull request 	 command mkdir ... #14 opened by joserra-15  1 linked pull request 
 "clear" command ... #5 opened by joserra-15	 Browser compatibility ... #30 opened by joserra-15
 function library(string,string) ... #2 opened by joserra-15 	 command mv ... #15 opened by joserra-15  1 linked pull request 
 the history of commands ... #4 opened by joserra-15  1 linked pull request 	 Responsive design ... #29 opened by joserra-15 
 Keyboard shortcuts ... #6 opened by joserra-15 1 linked pull request 	 command cat ... #13 opened by joserra-15 
 the "man" command ... #20 opened by joserra-15 	 command echo ... #16 opened by joserra-15 
 Additional commands ... #31 opened by joserra-15	 autocomplete with tab ... #24 opened by joserra-15  1 linked pull request 
 command ls and cd ... #3 opened by joserra-15 	 bugs ... #27 opened by joserra-15 1 linked pull request 
 command "JS" ... #21 opened by joserra-15 	

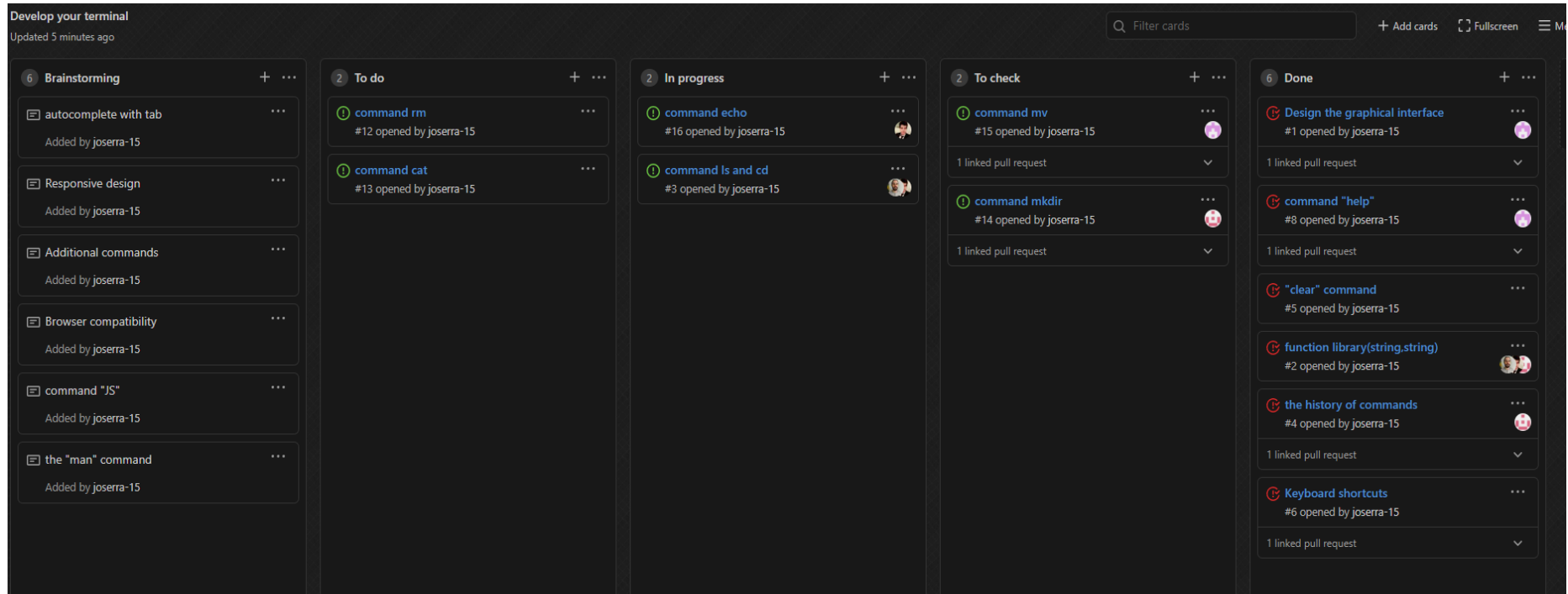
Anexo 2. Vista del tablero de procesos creado



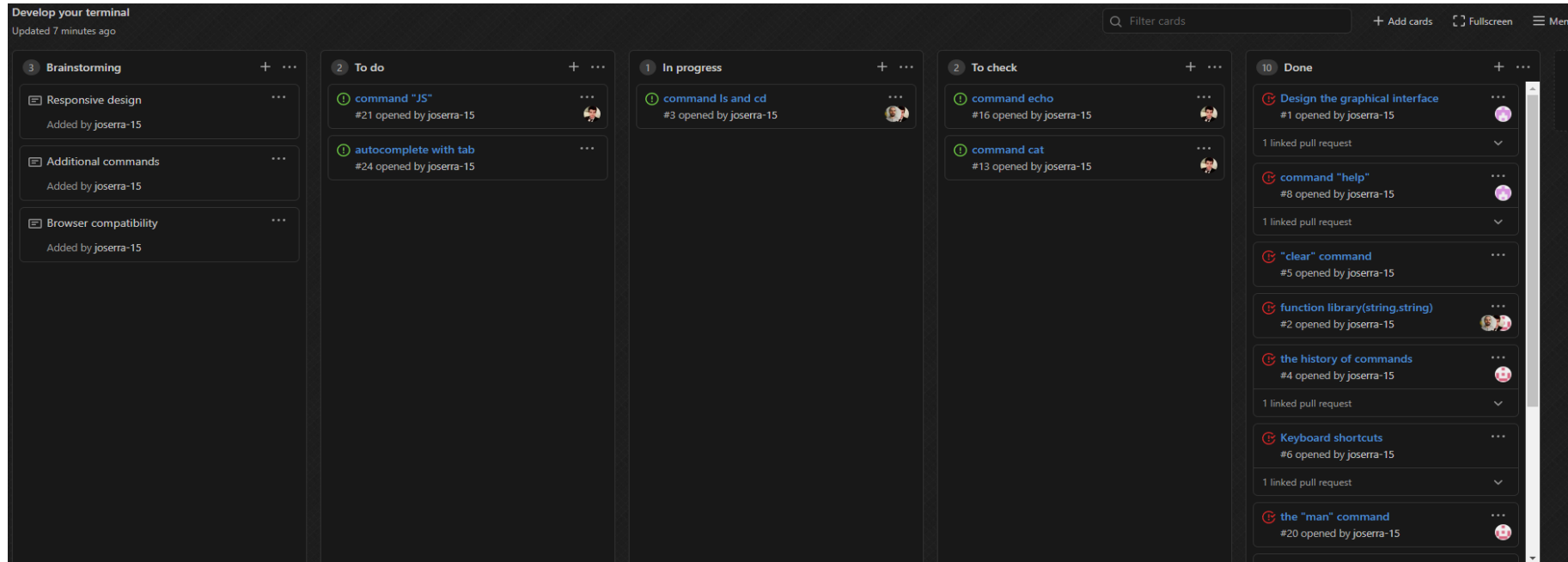
Anexo 3. Vista de la pizarra Kanban



Anexo 4. Evolución de las tareas



Anexo 5. Evolución de tareas



Anexo 6. Evolución de tareas

