Documentación API Poker Planner

Programas requeridos para el proyecto:

* GitHub Desktop para crear el repositorio y trabajar con él.
* Docker Desktop para tener una interfaz gráfica para manejar el contenedor.
* VsCode con la extensión VUE para programar.
* Postman para probar los endpoints.
* NodeJS para poder utilizar código JS en entorno servidor
* Se recomienda utilizar las últimas versiones disponibles para garantizar su estabilidad y funcionalidad.

Partimos de la base que la API se hace con Express, con respaldo de una BBDD MongoDB y que todo está contenerizado en Docker.

En el caso de ejecutar el proyecto por primera vez, es necesario componer el contenedor con Docker compose up - - build (los menos van juntos). Una vez hecho, aparecerá en Docker Desktop.

Para los primeros pasos con VUE es recomendable seguir los pasos descritos en [https://vuejs.org/guide/quick-start.html](%20https:/vuejs.org/guide/quick-start.html) en los que nos mostrará las configuraciones iniciales, dependencias y la forma de arrancarlo.

Se ha reutilizado la estructura ya realizada con el proyecto de React, con algunos cambios para empezar de nuevo.

Para ejecutar la API es necesario primero arrancar el contenedor, para después introducir mediante consola de comandos (en modo administrador) y en la ruta donde está el proyecto npm run dev con el fin de arrancar el entorno de desarrollador de VUE.

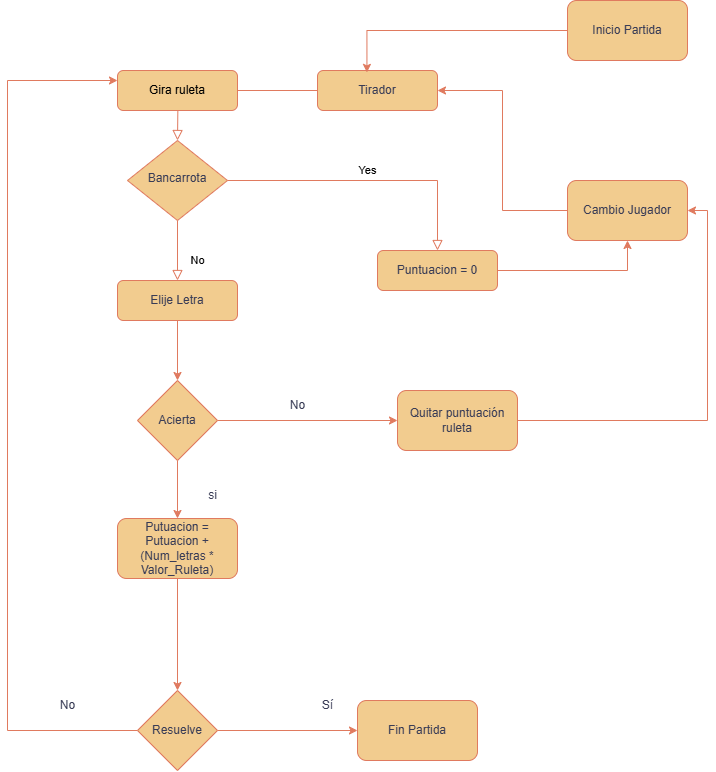
La dirección local a la que accedemos es la que nos devolverá la consola de comandos.  
  
En un principio entraremos en la página de login, mediante la cual iniciaremos sesión o nos registremos en el caso de no haberlo hecho (tiene una página dedicada para ello y no permite registrar a dos usuarios con el mismo email).  
  
Una vez se hace login, entraremos en una página landing la cual contiene accesos a 2 versiones del Ahorcado, un Poker Planner y una Ruleta de la Suerte. Se ha introducido una barra de navegación la cual nos muestra un botón de Inicio (que nos devuelve a esta misma página) y un botón para cerrar la sesión y que nos devuelva a la página de login (por implementar) y las fichas de las opciones se ha encajado en un grid para darle uniformidad y dinamismo.

En cada ficha hay una breve descripción de lo que consiste cada “mini aplicación”.

Se ha creado un componente NavBar en la cual mostraremos la barra de navegación en las páginas que lo necesiten.

Se ha creado un componente Modal en el cual se establecen los parámetros básicos para mostrar una ventana modal cuando ganamos y/o perdemos en el ahorcado. Se puede personalizar para adaptarlo a la página en la que estemos.

Diagrama de funcionamiento de la ruleta:



A la ruleta hay que implementarle la opción de comprar vocal siempre y cuando el jugador tenga puntos suficientes para comprarla y sea su turno. Igualmente, si se equivoca pasa el turno al siguiente jugador.

En cuanto a las cosas pendientes de implementar podemos enumerar las siguientes:

* Adaptar la BBDD para acoger las sesiones del Poker Planner, así como sus resultados, y poder guardar los resultados de las partidas en los minijuegos.
* En la barra de navegación, debería mostrar el nombre del usuario que se ha conectado. Para ello es necesario generar un token con el inicio de sesión y que el sistema lo compruebe para mostrar la información de forma adecuada.
* Hacer que el botón cerrar sesión de la barra de navegación haga lo propio y nos redirija a login.
* Los minijuegos del Ahorcado son versiones preliminares, se debe cambiar por las versiones definitivas, siguiendo la estructura de carpetas y modificando el archivo router.js de la carpeta public para gestionar las rutas de forma correcta y su posterior reflejo en el archivo landing.vue para que todo funcione correctamente.
* Poker Planner necesita no sólo su reconversión de la versión anterior en React a VUE, sino también de la construcción de la lógica del backend.
* Ruleta de la suerte necesita su conversión a VUE (mejor rehacerla), ya que en React estaba construida como un componente. Esta filosofía choca con VUE que es ávida de tener HTML, CSS y JS en el mismo fichero para su implementación.
* En ruleta es necesario que cuando se unan jugadores a la partida, se muestre un cuadro con su puntuación y nombre de usuario (manejar token de la sesión).
* En Poker Planner es necesario que Scrum Master tenga una barra de opciones específica para sus funciones (elegir máximo, elegir mínimo, repetir votación, elegir media y asignar tarea), mientras que el resto sólo tendrán opciones para votar.
* Es necesario crear los endpoints necesarios para acometer los procesos de autenticación, token de login, creación de tareas por Scrum Master, creación de salas para los minijuegos…
* Una vez que todo esté construido, es necesario hacer pruebas de concurrencia con el fin de garantizar que las sesiones funcionan de forma correcta y que los datos se guardan correctamente en la BBDD.
* Sería muy recomendable hacer una reestructuración de carpetas con el fin de organizar el proyecto de forma más óptima, actualizando las rutas de acceso en todos los archivos.