



PEEPOTYPE

PROYECTO FINAL - DAW

ANDREA SÁNCHEZ
DAW 2021/2022
TUTOR: SERGI GRAU

Contenido

1. RESUMEN INICIAL.....	2
2. ESTUDIO DE VIABILIDAD.....	3
2.1 ESTABLECIMIENTO DEL ABASTO DEL SISTEMA	3
2.2 ESTABLECIMIENTO DE LA SITUACIÓN ACTUAL	3
2.3 DEFINICIÓN DE LOS REQUISITOS DEL SISTEMA.....	4
2.4 ESTUDIO DE LAS ALTERNATIVAS DE SOLUCIÓN	4
2.5 VALORACIÓN DE LAS ALTERNATIVAS	4
2.6 SELECCIÓN DE LA SOLUCIÓN	4
3. ANALISIS DEL SISTEMA	5
3.1 DEFINICIÓN DEL SISTEMA	5
3.2 ESTABLECIMIENTO DE REQUISITOS	5
3.3 DEFINICIONES DE INTERFACES DE USUARIO.....	6
3.4 ESPECIFICACIÓN DEL PLANO DE PRUEBAS.....	6
4. DISEÑO DEL SISTEMA	7
4.0 GUIA DE ESTILOS, USABILIDAD, ACCESIBILIDAD	7
4.1 ARQUITECTURA WEB	9
4.1.1 DEFINICIÓN DE NIVELES DE ARQUITECTURA: FRONTEND Y BACKEND.....	9
4.1.2 ESPECIFICACIÓN DE ESTANDARES, NORMAS DE DISEÑO Y CONSTRUCCIÓN	9
4.2 REVISIÓN DE CASOS DE USO	10
4.2.1 REVISIÓN DE SUBSISTEMAS DEPENDIENDO DE LOS CASOS DE USO	10
4.2.2 REQUISITOS DE IMPLANTACIÓN	11
4.3 ANALISIS PARADIGMA ESTRUCTURADO / ORIENTADO A OBJETOS.....	12
4.4 PERSISTENCIA DE DATOS: ANALISIS Y DISEÑO	13
5. DESARROLLO	14
5.1 PLANIFICACIÓN DE LAS ACTIVIDADES DE DESARROLLO	14
5.2 DESARROLLO	14
5.3 DOCUMENTACIÓN TÉCNICA DEL PROGRAMARIO	14
6. IMPLANTACIÓN	15
6.1 FORMACIÓN	15
6.2 IMPLANTACIÓN DEL SISTEMA Y PRUEBAS	15
6.3 NIVEL DE SERVICIOS	15
6.4 ACEPTACIÓN DEL SISTEMA.....	15
7. MANTENIMIENTO Y VERSIONES FUTURAS	15

1. RESUMEN INICIAL

Se trata de una aplicación de mecanografía en la que puedes probar tu nivel, ya sea en solitario o contra alguien. La aplicación te mostrará el WPM, la precisión y los errores. En modo solitario hay diferentes modos, de 60 segundos a 10 segundos y otro modo que no puedes ver lo que escribes. Todos los datos se guardarán para que puedas verlos más adelante.

La aplicación ha sido realizada con JavaScript, ChartJS, EXPRESS, WebSockets y MongoDB también usando LocalStorage.

Es tracta d'una aplicació de mecanografia en la que pots provar el teu nivel, ja sigui en solitari o contra algú. La aplicació et mostrarà el WPM, la precisió i els errors. En mode solitari hi han diferents modes, de 60 segons a 10 segons i un altre en el que no pots veure el que escribeixes. Totes les dades es guardarán perquè puguis mirar-les més endavant.

La aplicació ha sigut realitzada amb JavaScript, ChartJS, EXPRESS, Websockets i MongoDB també usant LocalStorage.

It is a typing application where you can test your level, either alone or against someone. The app will show you the WPM, accuracy and errors. In solo mode there are different modes, from 60 seconds to 10 seconds and another mode that you can't see what you write. All data will be saved so you can view it later.

The application has been made with JavaScript, ChartJS, EXPRESS, WebSockets and MongoDB also using LocalStorage.

2. ESTUDIO DE VIABILIDAD

2.1 ESTABLECIMIENTO DEL ABASTO DEL SISTEMA

El proyecto PEEPOTYPE es FrontEnd y también tiene una parte de BackEnd. Los lenguajes que he usado para FrontEnd serían: ChartJS, que puedes usarlo sin pagar y sirve para crear diferentes tipos de tablas. Para Backend he usado: Websockets, para abrir una sesión de comunicación entre el usuario y el servidor(en este caso express), el cual puedes enviar y recibir mensajes al servidor y recibir respuestas por eventos. El servidor Express que está alojado dentro del entorno de NodeJS y es un framework, LocalStorage y MongoDB, del cual he utilizado jQuery y AJAX para poder hacer una conexión con el cliente, el servidor y Mongo.

2.2 ESTABLECIMIENTO DE LA SITUACIÓN ACTUAL

La idea del proyecto salió de una página en la que usaba para practicar mecanografía y ver mis resultados. Esa fue mi primera inspiración para hacer este proyecto, yo creo que es una buena idea porque todos pueden hacer servir la página.

Hay diferentes páginas, pero cada una se centra más en un tema en concreto, ya sea solo mejorar la mecanografía con diferentes cursos, o simplemente ver tus resultados, o para competir contra otra gente.

Mi página es una mezcla de todas las anteriores y eso hace que se diferencie de las otras páginas en el mercado actual.

Una aplicación de mecanografía a destacar del mercado sería *MonkeyType*, es la que en el primer párrafo me refería, tiene una idea muy definida y aunque solo se centre en el modo solitario tiene un aspecto visual muy minimalista y formal.

2. ESTUDIO DE VIABILIDAD

2.3 DEFINICIÓN DE LOS REQUISITOS DEL SISTEMA

Para que tu mismo puedas probar o desarrollar esta aplicación web deberás tener instalados los siguientes lenguajes: MongoDB, Express y Websockets.

En cambio para probar o usar la página web PEEPOTYPE no requerirás de ninguna, pero si que no se podrán los valores ya que no he tenido en cuenta poner una base de datos en la que todo el mundo pueda guardar datos en vez de localmente, si fuera el caso de que hubiera más versiones en el futuro, el primer cambio en la aplicación sería ese, y en vez de usar MongoDB usaría MongoDB Atlas.

2.4 ESTUDIO DE LAS ALTERNATIVAS DE SOLUCIÓN

2.5 VALORACIÓN DE LAS ALTERNATIVAS

2.6 SELECCIÓN DE LA SOLUCIÓN

El proyecto está programado con JavaScript usando EXPRESS y Websockets, además de MONGO y Local Storage, también aparte he usado ChartJS.

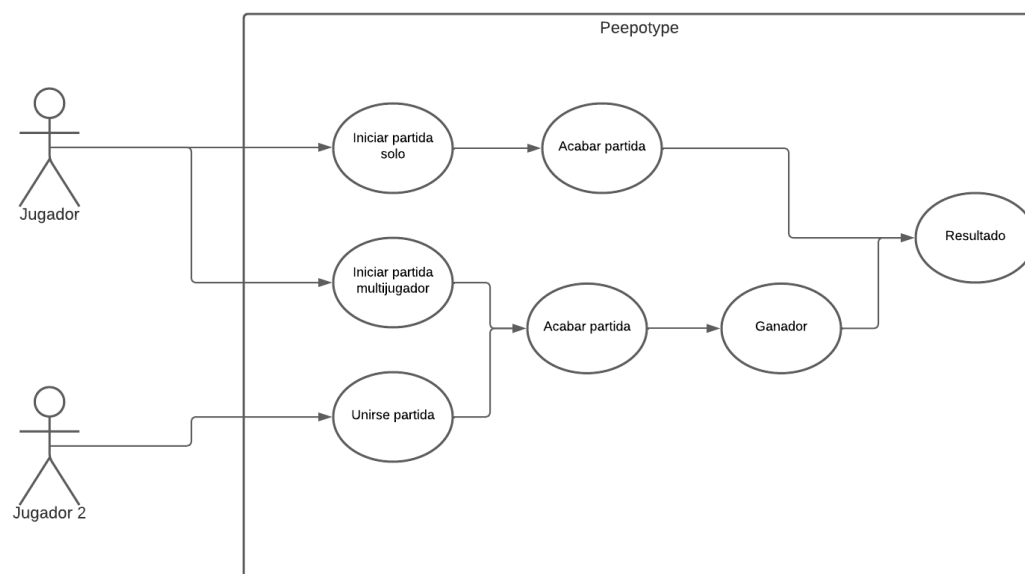
Tomé la decisión de usarlo con JavaScript fue porque tenía pensado usar WebSockets y no sabía si se me podría complicar usando otro entorno de trabajo. Claramente, pensé en desarrollarlo con otro como Angular pero finalmente me decidí por JavaScript, por la simplicidad. Aunque cabe decir que en algunos puntos hubiese sido mucho mejor haber usado Angular.

3. ANALISIS DEL SISTEMA

3.1 DEFINICIÓN DEL SISTEMA

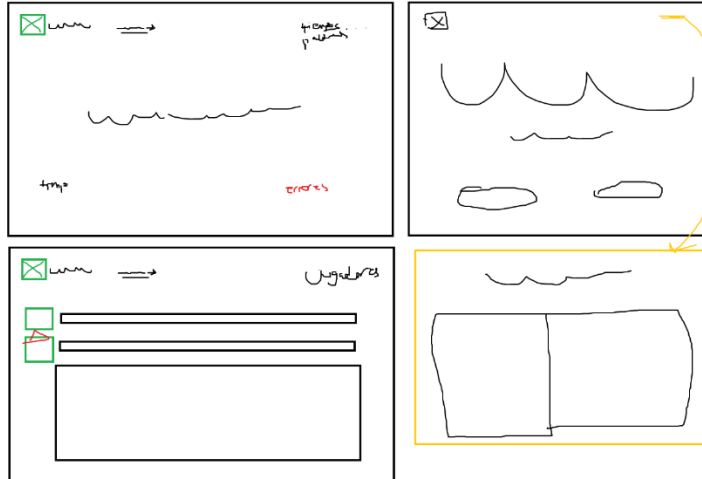
El proyecto sería más enfocado a FrontEnd aunque también tiene una base de datos con Mongo y LocalStorage bastante simple para poder guardar los resultados de los jugadores que jueguen, LocalStorage está enfocado para el modo solitario de un jugador el cual te mostrará todos los resultados de todas tus partidas y utilizo Mongo también en solitario, en cambio este servirá si quieres guardar tu puntuación con tu nombre. Los dos datos guardados se verán en la misma página.

3.2 ESTABLECIMIENTO DE REQUISITOS



3. ANALISIS DEL SISTEMA

3.3 DEFINICIONES DE INTERFACES DE USUARIO



Este ha sido el mockup que hice para la página, tenía pensado hacer 4 páginas con un inicio, otro para el solitario, multijugador y la puntuación. Viendo el proyecto final y el mockup creo que he sido fiel al diseño original.

3. ANALISIS DEL SISTEMA

3.4 ESPECIFICACIÓN DEL PLANO DE PRUEBAS

Para la realización de pruebas pedí a conocidos si podían darme una opinión de que les parecía mi página. Cada uno dio distintos puntos de vista o errores que vieron y con ello pude cambiar y mejorar el proyecto a como actualmente está.

4. DISEÑO DEL SISTEMA

4.0 GUIA DE ESTILOS, USABILIDAD, ACCESIBILIDAD

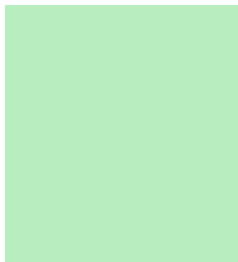
Logo:

La imagen representativa de Peepotype sería ese simple texto en mayúsculas, el cual es simple y llamativo. Por otra parte el logo que representaría en si, sería el icono.

PEEPOTYPE



Colores básicos de la marca:



#b7edbf

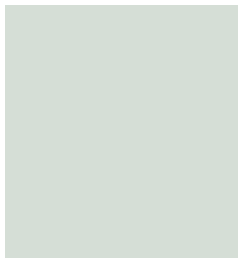
RGB
183, 237, 191

CMYK
23%, 0%, 19%, 7%

HSV
129°, 23%, 93%

HSL
129°, 60%, 82%

Colores complementarios:



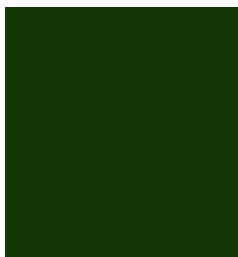
#d5ded6

RGB
213, 222, 214

CMYK
4%, 0%, 4%, 13%

HSV
127°, 4%, 87%

HSL
127°, 12%, 85%



#133805

RGB
19, 56, 5

CMYK
66%, 0%, 91%, 78%

HSV
104°, 91%, 22%

HSL
104°, 84%, 12%

Tipografías:

Escogí tipografías sin serifa, de palo seco.

- Monsterrat Italic:

ABCDEFGHIJKLM
NOPQRSTUVWXYZ

- DMS-Regular:

ABCDEFGHIJKLM
NOPQRSTUVWXYZ

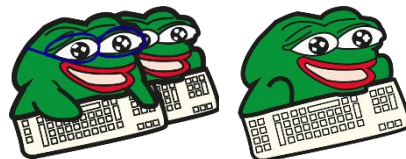
- Radio Canada:

ABCDEFGHIJKL
MNOPQRSTUVW
XYZ

Imágenes e iconos:

Todas las imágenes son .png, dos de los iconos son creados con illustrator.

Los dos iconos creados por mi:



- **ACCESIBILIDAD**

Una definición rápida de accesibilidad sería diseñar una página web teniendo en cuenta las personas con discapacidades y no influya negativamente su experiencia.

Cosas que he tenido en cuenta

Colores simples y que no molestan a la vista del usuario.

Maquetación lineal y lógica, entender la página web es muy simple y está bien estructurada para ver dónde ir en todo momento.

4. DISEÑO DEL SISTEMA

4.1 ARQUITECTURA WEB

4.1.1 DEFINICIÓN DE NIVELES DE ARQUITECTURA: FRONTEND Y BACKEND

Utilice como bien he ido comentando, JavaScript con una parte servidor de Express para poder hacer llamadas al servidor con el cliente, eso era necesario tanto para WebSockets como para MongoDB, el cual para Mongo hago llamadas con jQuery y Ajax, ya bien para guardar los valores con un POST o con un GET.

Para que funcionará bien todo en la parte del servidor, todos los archivos .js, .json y css están en una carpeta llamada public que luego en el server llamo.

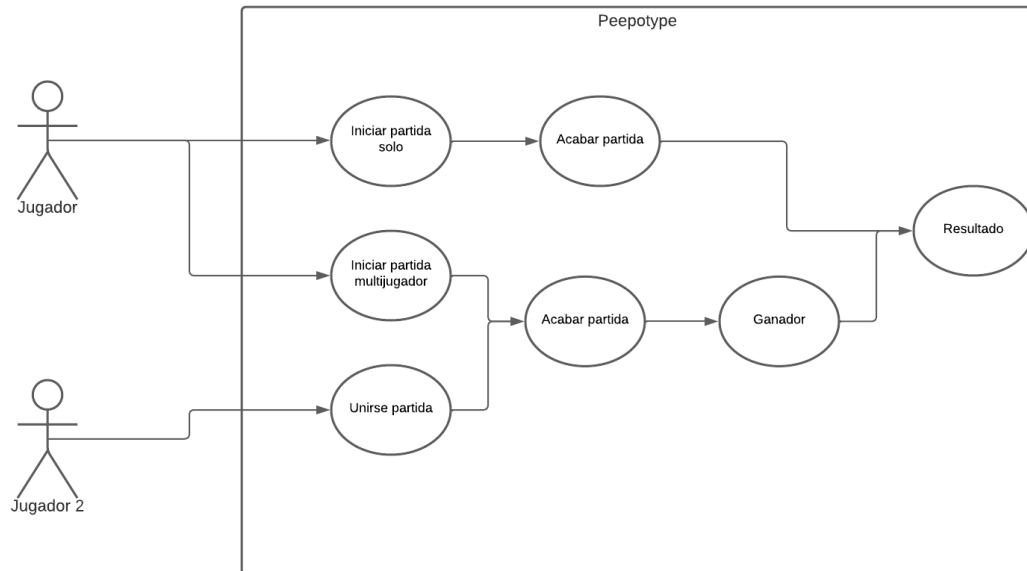
4.1.2 ESPECIFICACIÓN DE ESTANDARES, NORMAS DE DISEÑO Y CONSTRUCCIÓN

Mi proyecto no está pensado para que cueste dinero en un futuro o se requiera de una suscripción, será todo disponible para el usuario sin ningún coste. No pensé en añadir ninguna interfaz en la que requiera el usuario tener privilegios pagando entonces por eso decidí que todo estuviera al abasto de cualquier persona que use la aplicación.

4. DISEÑO DEL SISTEMA

4.2 REVISIÓN DE CASOS DE USO

4.2.1 REVISIÓN DE SUBSISTEMAS DEPENDIENDO DE LOS CASOS DE USO



Una explicación más detallada sobre el diagrama de casos de uso que he hecho sería que está hecho con actores y tiene el recuadro grande (sistema) con el nombre de PeepoType porque esa sería la página web en la que los actores entran, allá ya se encuentran los casos de uso que me he centrado en cuando empieza la partida (ya sea en solitario o multijugador) y cuando la acaba que lleva a un mismo lado, el resultado.

4. DISEÑO DEL SISTEMA

4.2 REVISIÓN DE CASOS DE USO

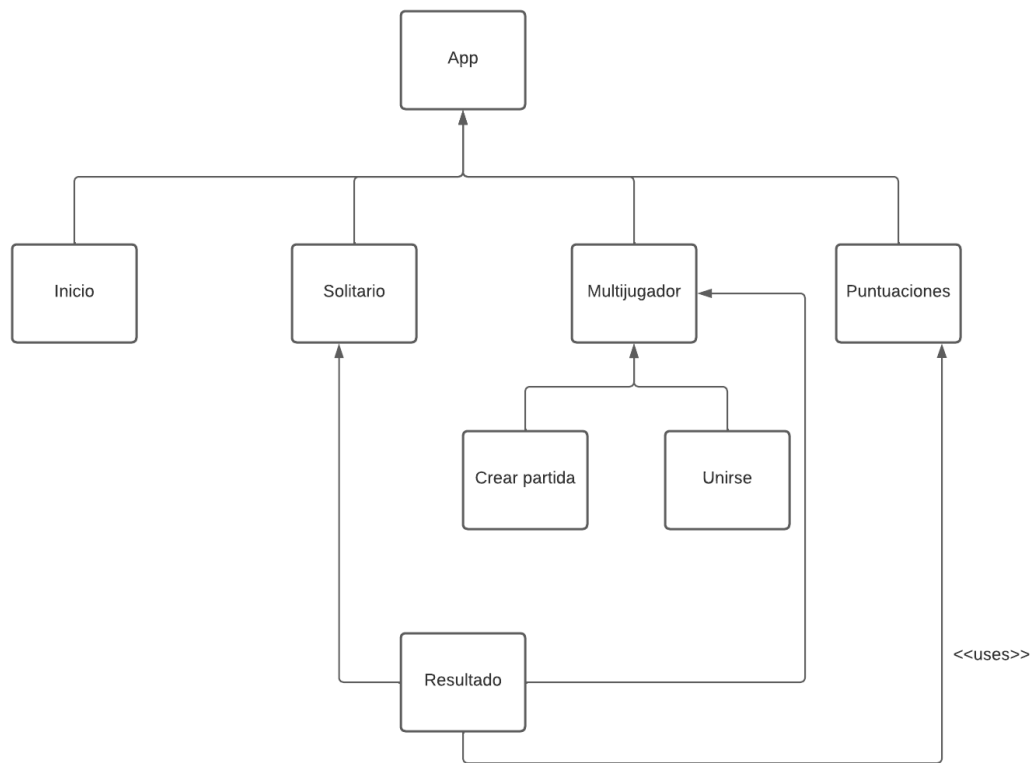
4.2.2 REQUISITOS DE IMPLANTACIÓN

Para que la aplicación funcione para todos los usuarios se debería de deployar con una URL propia, ya sea gratuita o de pago, yo pensé en usar Heroku ya que puedes usar una versión sin pagar y sirve como servicio PaaS y allá puedas usar tu aplicación.

En mi caso, mi aplicación no estaría lista para implementar porque faltaría cambiar un poco de código y rehacerlo con Mongo Atlas que es compatible con Heroku y servirá como base de datos, hasta ahora como proyecto inicial usaba mi propio MongoDB de forma local y si subiera esta aplicación web la gente no podría guardar sus puntuaciones.

4. DISEÑO DEL SISTEMA

4.3 ANALISIS PARADIGMA ESTRUCTURADO / ORIENTADO A OBJETOS



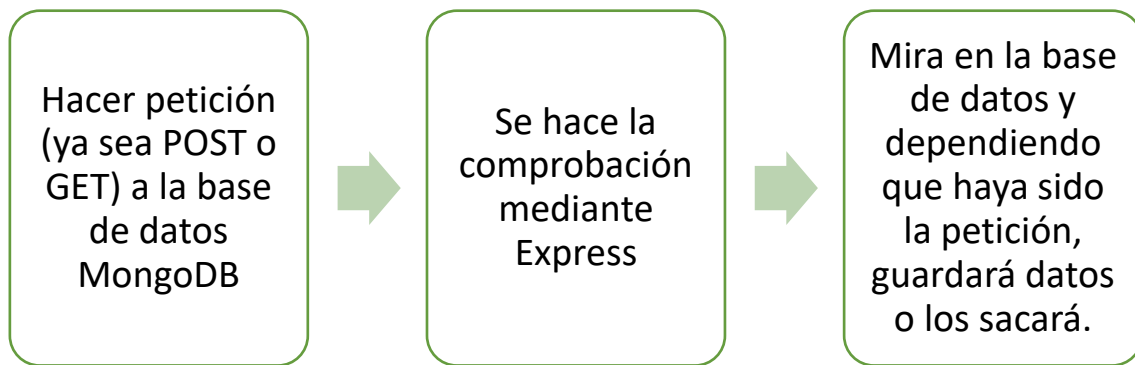
Está sería un diagrama de clases simple sin relaciones ni multiplicidades. Son clases que creo que son importantes y destacan las cuales van con la App.

4. DISEÑO DEL SISTEMA

4.4 PERSISTENCIA DE DATOS: ANALISIS Y DISEÑO

El usuario puede guardar o recoger valores guardados en LocalStorage o Mongo.

Para MongoDB se debería hacer como el diagrama de abajo demuestra.



5. DESAROLLO

5.1 PLANIFICACIÓN DE LAS ACTIVIDADES DE DESARROLLO

Primera semana	Segunda semana	Tercera semana
<ul style="list-style-type: none">•Comenzar proyecto•Elegir lenguaje e ir creando los archivos. Empezar a hacer Mockup y diseño visual de la página con HTML5 y CSS.	<ul style="list-style-type: none">•Empezar a realizar el juego del modo solitario con distintos modos y guardarlos en ChartJS•Guardar las puntuaciones finales en LocalStorage.•Tener todo el servidor hecho de EXPRESS	<ul style="list-style-type: none">•Realización del multijugador con WebSockets y mostrar el resultado de las dos personas.•Guardar puntuación del modo solitario si el usuario quiere con nombre en MongoDB y enseñarlo.•Realización de UML de casos de uso y memoria con vídeo demostrativo.



5.2 DESAROLLO

Para ver como he ido desarrollando la página puedes consultar en GitHub:

<https://github.com/ansasa2122daw2/PROYECTOFINAL>

Si no también puedes mirar el fichero adjunto en la entrega para ver el resultado final.

5.3 DOCUMENTACIÓN TÉCNICA DEL PROGRAMARIO

Toda la documentación y explicación detallada sobre este está dentro de los archivos JS propios, donde dejo apuntes e aclaraciones para poder tener todo más ordenado y conciso.

6. IMPLANTACIÓN

6.1 FORMACIÓN

No requieres de ninguna formación ni documentación inicial para poder entender o usar la aplicación ya que se ha usado lenguajes sencillos y del abasto de cualquier persona. Toda información puede ser encontrada fácilmente en Internet. Tampoco tiene ningún sistema de administración por lo cual no necesitas información extra sobre eso.

6.2 IMPLANTACIÓN DEL SISTEMA Y PRUEBAS

Se ha testeado con console logs para poder comprobar el error, también poniendo debuggers en el código y con ello se han podido solucionar correctamente.

6.3 NIVEL DE SERVICIOS

No hay ningún sistema de suscripción, demo o pago para esta aplicación web, todo es gratuito y está disponible a todos los públicos sin tener que aportar ningún tipo de servicio monetario.

6.4 ACEPTACIÓN DEL SISTEMA

No ha habido ningún problema con el cliente entendiendo la aplicación lo que puede que haya sido confusa la parte del multijugador, esa misma, si la aplicación se vuelve más novedosa y famosa se podría mirar a mejorarla, como otros ámbitos de la página web.

7. MANTENIMIENTO Y VERSIONES FUTURAS

Alguna idea del futuro sería que se pudiera usar una URL para que puedan probarlo todo tipo de personas el modo multijugador ya que sigue en “desarrollo” y las puntuaciones se puedan guardar para cualquier persona que lo use. Para ello se necesitaría unas versiones futuras.