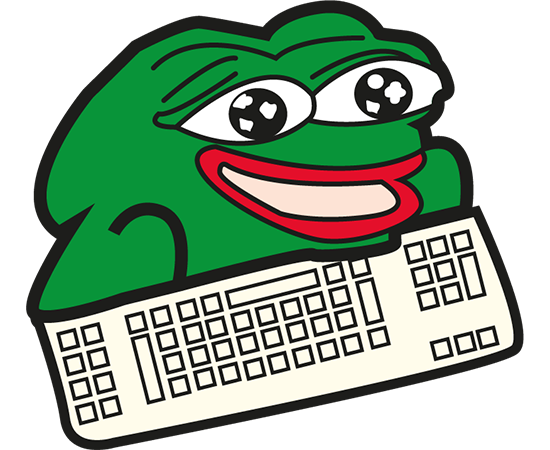
*PEEPOTYPE*



Andrea Sánchez

DAW2

Tutor: Sergi Grau

**INDEX**

1. Resumen inicial ………………………………………………………………………………………………………………… 3

2. Estudio de viabilidad

2.1. Estudio de la situación actual …………………………………………………………………………… 6

2.2. Definición de los requisitos del sistema ……………………………………………………….…… 5

2.3. Estudio de las alternativas de solución y valoración de las alternativas ……….…... 4

3. Análisis del sistema

3.1. Definición del sistema ………………………………………………………………..……………………. 6

3.2. Establecimiento de los requisitos ………………………………………………..….……………….. 7

3.3. Definiciones de interfaces de usuario ……………………………………………..………………… 8

3.4. Especificación del plano de pruebas ………………………………………………….……….…….. 9

4. Diseño del sistema

4.0. Guía de estilos, accesibilidad …………………..………………………………….….……………….. 10

4.1. Arquitectura Web

4.1.1. Definición de niveles de arquitectura: FrontEnd y BackEnd ………….…. 14

4.1.2. Especificación de estándares, normes de diseño y construcción ………

4.2. Revisión de casos de uso ……………………………………………………………………………..….

4.2.2. Elección de alternatives de componentes i licencias más adecuadas .

4.2.3. Especificaciones de desarrollo y pruebas …………………………………………

4.5. Persistencia de datos: Análisis y diseño de bases de datos /ORM …………………...

5. Desarrollo.

5.1. Planificación de las actividades de desarrollo y integración del sistema ………….

6. Implantación.

6.1. Formación ………………………………………………………………………………………………………

6.2. Implantación del sistema i pruebas ………………………………………………………………..

6.3. Nivel de servicios ……………………………………………………………………………………………

6.4. Aceptación del sistema ………………………………………………………………………………….

7. Mantenimiento i versiones futures ……………………………………………………………………………….

**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 aplicación de mecanografía en la que pots probar el teu nivel, ja sigui en solitari o contra algú. La aplicación et mostrará 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 aplicación ha sigut realitzada amb JavaScript, ChartJS, EXPRESS, Websockets i MongoDB també usant LocalStorage.

**2. ESTUDIO DE VIABILIDAD**

**2.1 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 diferencies 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**

**2.2 DEFINICIÓN DE LOS REQUISITIOS DEL SISTEMA**

Para la definición de los requisitos del sistema he creado un UML de casos de uso, el cual uso el diagrama de actores.

**2. ESTUDIO DE VIABILIDAD**

**2.3 ESTUDIO DE LAS ALTERNATIVAS DE SOLUCIÓN Y VALORACIÓN DE LAS ALTERNATIVAS**

El proyecto está programado con JavaScript usando EXPRESS y Websockets, además de MONGO y Local Storage.

Tomé la decisión de usarlo con JavaScript fue porque tenía pensado usar WebSockets y no sabia si se me podría complicar usando otro entorno de trabajo. Claramente, pensé en desenvoluparlo con otro como Angular pero finalmente me decidí por JavaScript.

**3. ANALISIS DEL SISTEMA**

**3.1 DEFINICIÓN DEL SISTEMA**

Estructura de la aplicación:

**3. ANALISIS DEL SISTEMA**

**3.2 ESTABLECIMIENTO DE REQUISITOS**

RQ001 – El usuario puede jugar una partida en solitario

RQ002 – El usuario puede ver sus puntuaciones al acabar la partida

RQ003 – El usuario puede ver todas sus anteriores puntuaciones

RQ004 – El usuario puede competir con otra persona

RQ005 – El usuario puede comparar sus resultados con el oponente

RQ006 – El usuario en una partida puede guardar su nombre y ver sus resultados

**3. ANALISIS DEL SISTEMA**

**3.3 DEFINICIONES DE INTERFACES DE USUARIO**

**3. ANALISIS DEL SISTEMA**

**3.4 ESPECIFICACIÓN DEL PLANO DE PRUEBAS**

**4. DISEÑO DEL SISTEMA**

**4.1 GUIA DE ESTILOS**

**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.

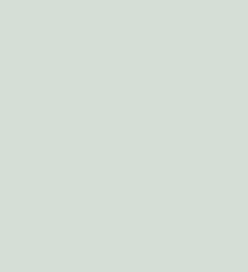
**Colores básicos de la marca:**

Imatge que conté plaça

Descripció generada automàticament #b7edbf



**Colores complementarios:**

 #d5ded6

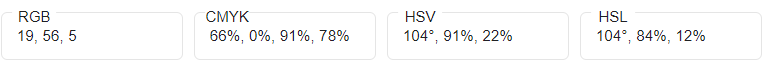


Imatge que conté plaça

Descripció generada automàticament #b7edbf



 #133805



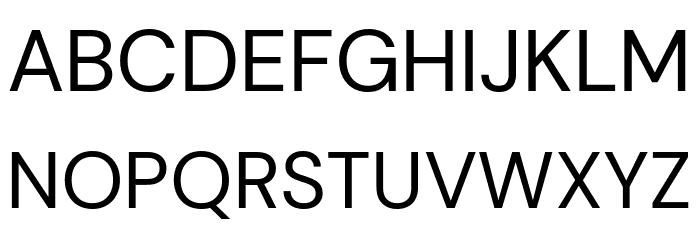
**Tipografías:**

Escogí tipografías sin serifa, de palo seco.

* Monsterrat Italic:



* DMS-Regular:



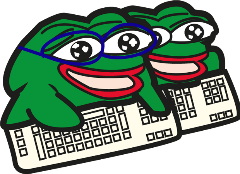
* Radio Canada:

Imatge que conté text

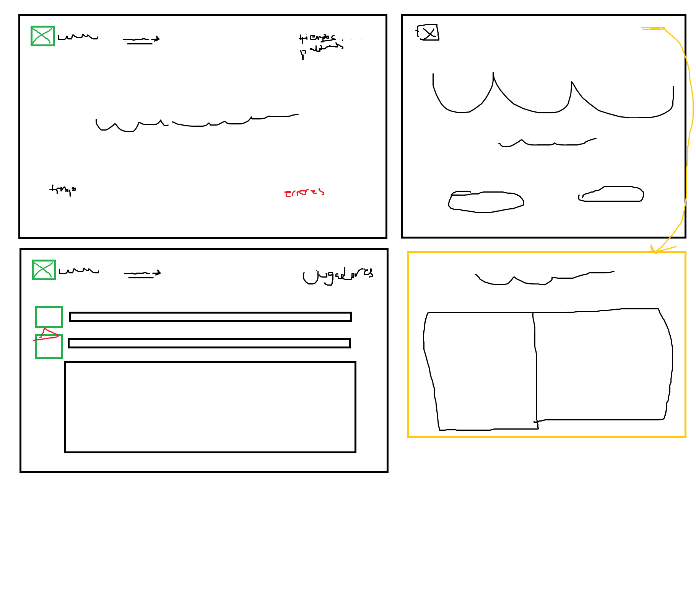
Descripció generada automàticament

**Imágenes e iconos:**

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

Los dos iconos creados por mi:

**Mockup:**

La siguiente imagen sería el mockup que hice nada más empezar el proyecto, el cual tenia pensado estructurarlo con 4 páginas, una para un jugador, los resultados, las puntuaciones y aparte también la página del multijugador. Actualmente está diferente estructurado.

Tenemos un índice que puedes navegar entre un solo jugador, multijugador y las puntuaciones.

**4.2 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 donde ir en todo momento.