*PROYECTO***:**

**Wordlabras**

|  |
| --- |
|  |

*Realizado por:* Adrián López Pastor

|  |  |
| --- | --- |
| http://alba9.es/imagenes/logo_galileo_1.jpg | **Desarrollo de Aplicaciones Web (DAW)** |
| **Valladolid, a 21 de** **Diciembre de 2022** |
| Nota: |  |

Índice

[1 Introducción 6](#_Toc121846771)

[1.1 Objeto de proyecto 6](#_Toc121846772)

[1.1.1 Breve Introducción 6](#_Toc121846773)

[1.1.2 Justificación 6](#_Toc121846774)

[1.1.3 Sentido del proyecto 7](#_Toc121846775)

[1.2 Lenguajes empleados 7](#_Toc121846776)

[1.2.1 Lenguajes 7](#_Toc121846777)

[1.2.2 IDEs 8](#_Toc121846778)

[1.3 Distribución 9](#_Toc121846779)

[1.3.1 Despliegue 9](#_Toc121846780)

[1.4 Requisitos de los clientes: 9](#_Toc121846781)

[1.5 Licenciamiento 9](#_Toc121846782)

[2 Recursos 10](#_Toc121846783)

[2.1 Hardware 10](#_Toc121846784)

[2.1.1 Para su desarrollo, Implementación y distribución 10](#_Toc121846785)

[2.2 Software 10](#_Toc121846786)

[2.2.1 Para su desarrollo, implementación y distribución 10](#_Toc121846787)

[2.3 Humanos 11](#_Toc121846788)

[2.4 Previsión económica del coste del proyecto 11](#_Toc121846789)

[2.4.1 Personal (pago por mano de obra) 11](#_Toc121846790)

[2.4.2 Coste de mantenimiento 12](#_Toc121846791)

[3 Descripción de la aplicación 12](#_Toc121846792)

[3.1 Funcionamiento general 12](#_Toc121846793)

[3.1.1 Diagrama de casos de uso 13](#_Toc121846794)

[3.1.2 Diagrama de secuencia 13](#_Toc121846795)

[3.2 Arquitectura 14](#_Toc121846796)

[3.2.1 Diseño de las bases de datos 14](#_Toc121846797)

[3.2.2 Arquitectura del sistema 15](#_Toc121846798)

[3.3 Interfaz 17](#_Toc121846799)

[3.3.1 Características generales 17](#_Toc121846800)

[3.3.2 Adaptación a dispositivos móviles 22](#_Toc121846801)

[3.3.3 Usabilidad/accesibilidad 26](#_Toc121846802)

[4 Autoevaluación y conclusiones 27](#_Toc121846803)

[4.1 Valoración del trabajo y dificultades encontradas 27](#_Toc121846804)

[4.2 Valoración de la herramienta o aplicación desarrollada 28](#_Toc121846805)

[4.3 Conclusiones finales 28](#_Toc121846806)

[4.3.1 Del diseño 28](#_Toc121846807)

[4.3.2 De la aplicación 28](#_Toc121846808)

[4.3.3 Del tiempo empleado 29](#_Toc121846809)

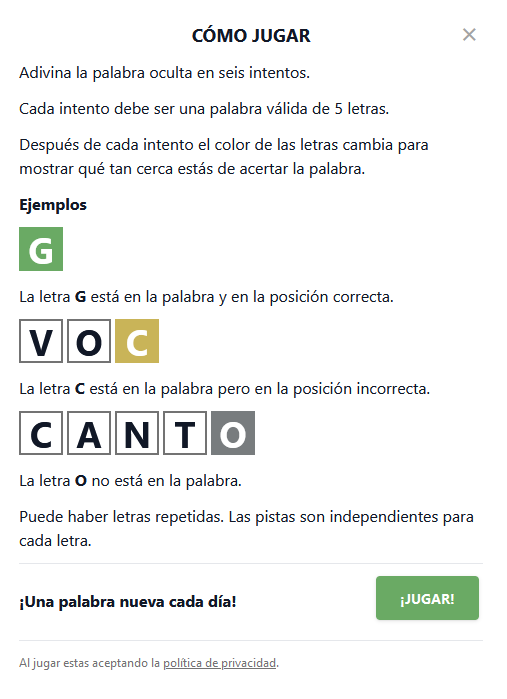
[4.3.4 De los conocimiendos adquiridos 29](#_Toc121846810)

[5 Bibliografía 29](#_Toc121846811)

# Introducción

## Objeto de proyecto

### Breve Introducción

El proyecto consiste en realizar una página web como el juego llamado “Wordle” que se ha hecho viral desde hace poco tiempo (en el tiempo en el que presento esta solicitud) de la página [https://wordle.danielfrg.com/](https://wordle.danielfrg.com/%20%20)  Dicho juego, se basa en tener una palabra de 5 letras diaria (cambia cada 24h) para todas las personas, esas personas tienen que adivinar esa palabra en 6 intentos. Está como aplicaciones y página web con muchas variantes (por ejemplo: Creación de retos, pones la palabra que quieran adivinar, el número de intentos y crea automáticamente la plantilla en el enlace que tienes que copiar y enviar a las personas que quieran. También existe en diferentes idiomas, de diferentes temáticas, por ejemplo, de Pokémon, etc…)

### Justificación

Realizo este proyecto por lo viral que se hizo/se ha hecho y lo divertido que es (al final es como un juego y es educativo).   
  
La innovación en este proyecto será contar con una base de datos en la que guardar/almacenar, gestionar y obtener los usuarios y con ello sus datos y estadísticas. Además, gracias a contar con una base de datos con los usuarios y sus resultados, haré un ranking de los mejores usuarios y su posición global respecto al resto de usuarios.

### Sentido del proyecto

Como he mencionado, ya existe una página web de éxito en el mercado que es en la que me he basado, aunque a partir de ahí hayan aparecido muchas más y lo realizo porque me parece divertido y entretenido de hacer.

## Lenguajes empleados

### Lenguajes

* HTML
  + HTML (Lenguaje de Marcas de Hipertexto, del inglés HyperText Markup Language) es el componente más básico de la Web. Define el significado y la estructura del contenido web. Además de HTML, generalmente se utilizan otras tecnologías para describir la apariencia/presentación de una página web (CSS) o la funcionalidad/comportamiento (JavaScript).[[1]](#footnote-1)
* CSS
  + Hojas de Estilo en Cascada (del inglés Cascading Style Sheets) o CSS es el lenguaje de estilos utilizado para describir la presentación de documentos HTML o XML (en-US) (incluyendo varios languages basados en XML como SVG, MathML o XHTML). CSS describe como debe ser renderizado el elemento estructurado en la pantalla, en papel, en el habla o en otros medios.[[2]](#footnote-2)
* JAVASCRIPT
  + JavaScript (JS) es un lenguaje de programación ligero, interpretado, o compilado justo-a-tiempo (just-in-time) con funciones de primera clase. Si bien es más conocido como un lenguaje de scripting (secuencias de comandos) para páginas web, y es usado en muchos entornos fuera del navegador, tal como Node.js, Apache CouchDB y Adobe Acrobat JavaScript es un lenguaje de programación basada en prototipos, multiparadigma, de un solo hilo, dinámico, con soporte para programación orientada a objetos, imperativa y declarativa (por ejemplo, programación funcional). Lee más en acerca de JavaScript.[[3]](#footnote-3)
  + Librerías:
    - Node.js
    - Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor basado en el lenguaje de programación JavaScript, asíncrono, con E/S de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google[[4]](#footnote-4)
    - Axios (utilizado para la conexión con la BBDD).
    - Axios es un Cliente HTTP basado en promesas para node.js y el navegador. Es isomorfico (= puede ejecutarse en el navegador y nodejs con el mismo código base). En el lado del servidor usa el modulo nativo http de node.js, mientras que en el lado del cliente (navegador) usa XMLHttpRequests.[[5]](#footnote-5)
* SQL
  + SQL es un lenguaje de dominio específico, diseñado para administrar, y recuperar información de sistemas de gestión de bases de datos relacionales.[[6]](#footnote-6)
  + Utilizando MySQL
    - MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos[[7]](#footnote-7)

### IDEs

* VisualStudio Code
* Phpmyadmin
  + Xampp

## Distribución

### Despliegue

Se necesitará una página de hosteo para poder jugar o en su defecto contar con todos los archivos necesarios a nivel local (Xampp + BBDD + Extensión de liveserver de VisualStudioCode).

## Requisitos de los clientes:

* Utilizar cualquiera de los siguientes navegadores y estas versiones o superiores (probablemente con versiones inferiores también se visualice correctamente hasta una versión no muy antigua)
  + **Firefox (Navegador preferente):** 107.0.1 (64 bit)
  + **Opera:** 92.0.4561.34
  + **Microsoft Edge:** Versión 108.0.1462.42 (Compilación oficial) (64 bits)
  + **Internet Explorer:** 
* Crearse un usuario a través de la página web
* Tener los requisitos mínimos de hardware para el posible acceso a la aplicación a través de la web.

## Licenciamiento

La licencia del proyecto será: “Attribution 4.0 International (CC BY 4.0)” que consiste en:

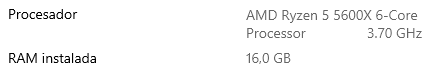
* Compartir - copiar y redistribuir el material en cualquier medio o formato
* Adaptar - remezclar, transformar y construir sobre el material para cualquier propósito, incluso comercial.

# Recursos

## Hardware

### Para su desarrollo, Implementación y distribución

Prestaciones del ordenador utilizado para el desarrollo:



1 TB de Espacio en el disco duro

## Software

### Para su desarrollo, implementación y distribución

* VisualStudio Code [IDE] [[8]](#footnote-8)
* Node.js
* Xampp [[9]](#footnote-9)

## Humanos

El número de horas totales empleadas para el proyecto es de:

* 64h-102h

De las cuales están divididas de la siguiente manera:

|  |  |
| --- | --- |
| **Informarse / Investigación** | 4h |
| **Memoria descriptiva** | 24h-32h |
| **Manual de código** | 8h-14h |
| **Programar / codificar** | 24h-40h |
| **Extras / Otros documentos** | 4h-12h |
| **Total:** | 64h-102h |

## Previsión económica del coste del proyecto

### Personal (pago por mano de obra)

El salario por hora de los empleados es de 8€/h por lo que si una persona realiza el proyecto sola como es en este caso, cobrará entre 512€ hasta 816€.

### Coste de mantenimiento

El hosteo de una página web será de unos 12€ anuales aproximadamente pudiendo ser el primer año gratuito en algunas páginas web. También se puede utilizar el hosting gratuito de herramientas y páginas como github o 000webhost.

En el caso de que el proyecto no necesite hosteo y sea utilizado de forma local, no tendrá coste alguno de mantenimiento.

# Descripción de la aplicación

## Funcionamiento general

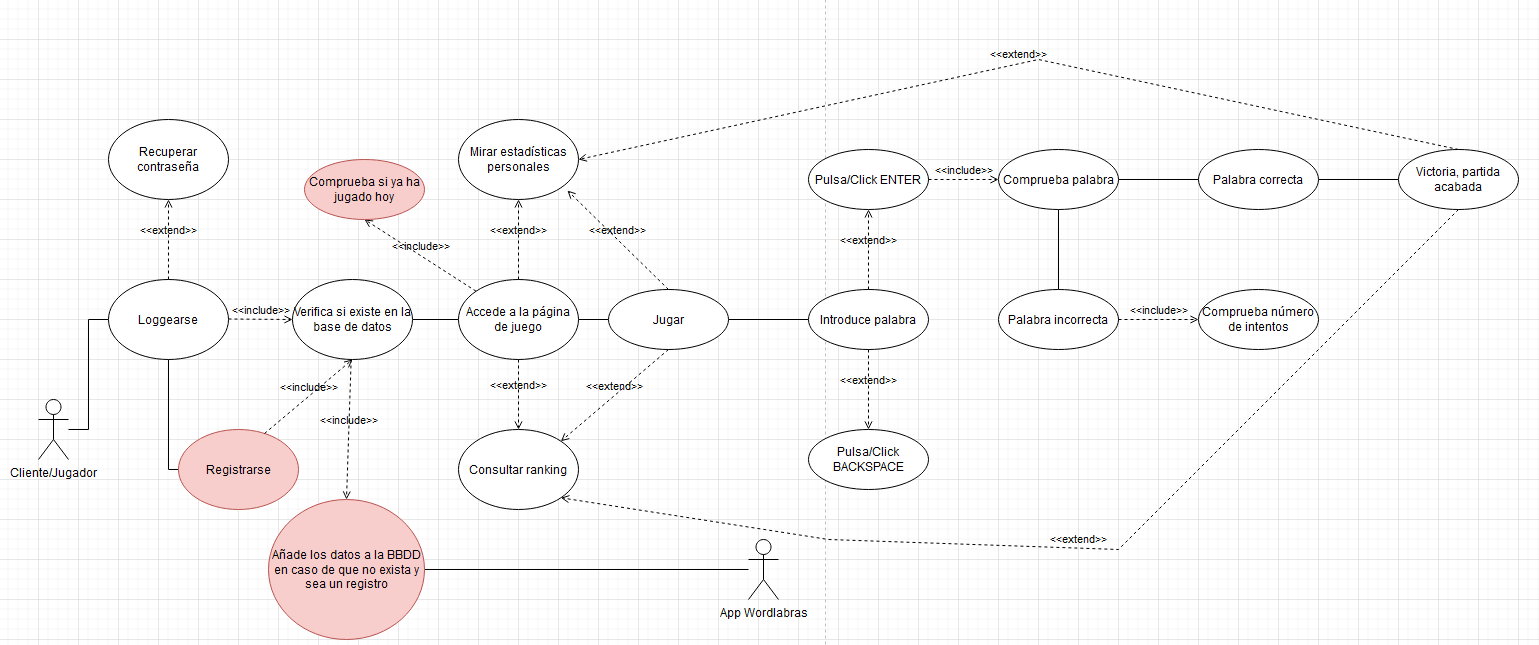
La app es una página web para poder jugar a algo similar al Wordle pero contando con un ranking y una base de datos. Al entrar en la página web. Tienes la opción de loggearte, de recuperar la contraseña olvidada y de registrarte (ésta última opción no estará implementada en el código al igual que todo lo rojo en los diagramas).

El registro funcionaría a través de un email, usuario y contraseña, comprobaría los datos si están en la base de datos y dependiendo de si está o no, el registro es exitoso o pide datos nuevos.

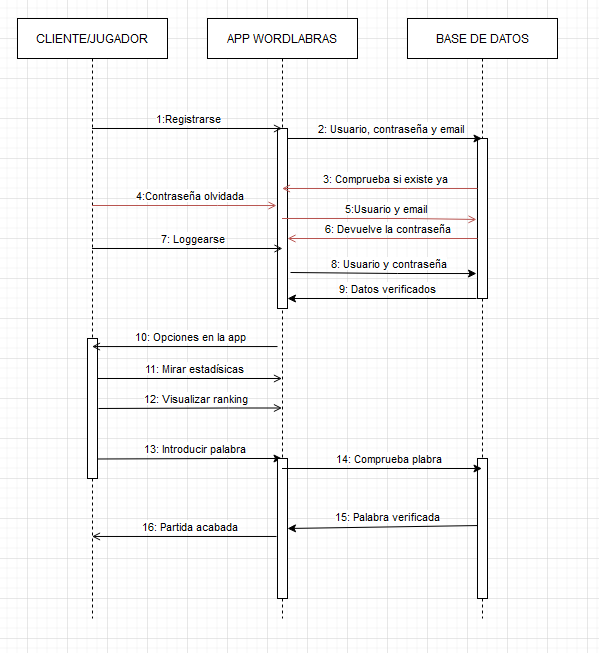
La opción de contraseña olvidada estaría de una manera más segura incluyendo el usuario y el email del usuario para obtener la contraseña, pero actualmente sólo contará con la necesidad del nombre de usuario.

Al loggearse correctamente muestra la ventana del juego con opciones para ver el ranking o las estadísticas del usuario. Cuando escribe una palabra, al pulsar ENTER comprueba si existe esa palabra, si la palabra no existe, informa al usuario y debe cambiarla hasta que formule una palabra que sí que existe. Tiene 6 intentos para adivinar la palabra del día

### Diagrama de casos de uso



### Diagrama de secuencia

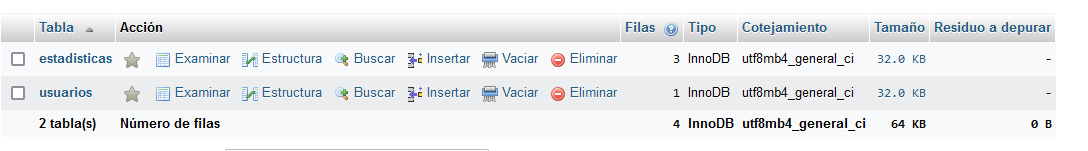


## Arquitectura

### Diseño de las bases de datos

#### Base de datos Wordlabras

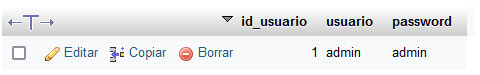
La base de datos se llama como el proyecto; “Wordlabras”, cuenta con una tabla de usuarios, otra de estadísticas.

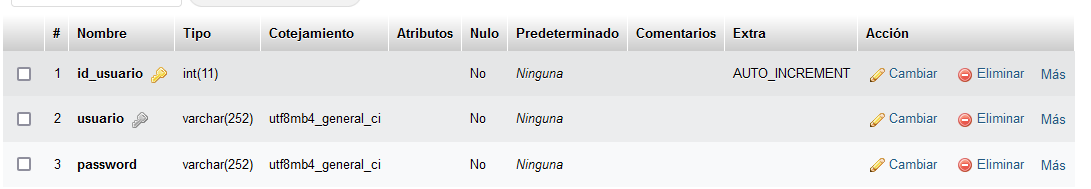


##### Tabla usuarios

La tabla usuarios cuenta con un id\_usuario (inservible, pero aporta información en caso de necesitar una mejora en el futuro), que no viene mal nunca tener un id único ajeno al usuario con el que poder trabajar de forma interna. Un usuario que tendrá un máximo de 252 caracteres y una contraseña que tendrá una cantidad máxima de 252 caracteres. El string de ‘usuario’ será único (si se implementase el registro, el campo único sería probablemente el email o ambos, email y usuario para mayor seguridad). La contraseña puede ser añadida a través del encriptado MD5 para mayor seguridad en la base de datos.

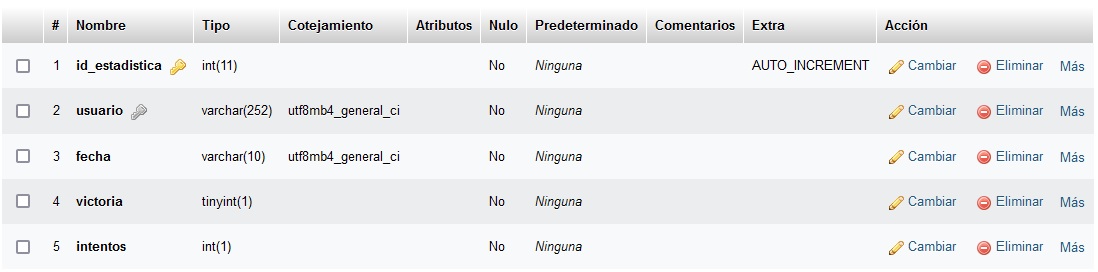
###### Campos de ejemplo





##### Tablas estadísticas

La tabla estadísticas cuenta con un id\_estadistica (inservible, pero aporta información en caso de necesitar una mejora en el futuro), que no viene mal nunca tener un id único ajeno al usuario con el que poder trabajar de forma interna. El usuario es una clave foránea cogida de la tabla usuarios gracias a que es un atributo único en la tabla usuarios. La fecha indicará el día que se jugó y los datos de ese día. La victoria es un tinyint que dirá si es true (1 a 255) o false (0) y dirá si obtuvo ese día la victoria o no. Los intentos es el número de intentos que tuvo ese día a la hora de ganar o perder (con un máximo de 6 intentos).



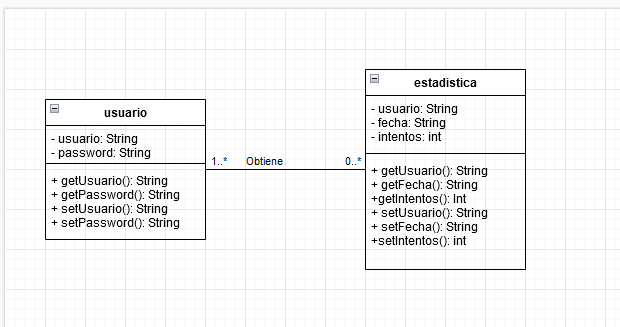
###### Campos de ejemplo



### Arquitectura del sistema

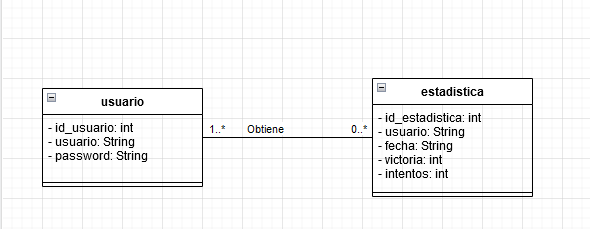
#### Diagrama de clases

En todo el código realizado sólo contaré con dos clases que serán útiles para el apartado de la base de datos, ya que creo que lo más cómodo para este caso será la utilización de las mismas.



#### Diagrama de objetos

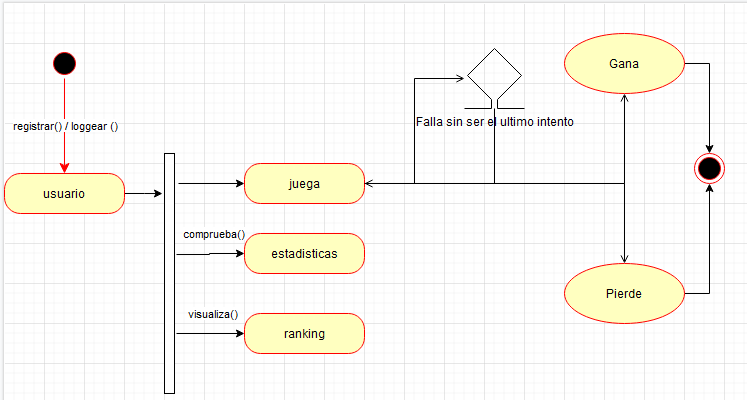
El diagrama de objetos es bastante similar al de clases al ser una relación sencilla



#### Diagrama de de estados de las clases principales

Empieza con un usuario que se loggea aunque también se puede registrar (no implementado, por eso no lo he desarrollado más en el diagrama). Una vez loggeado, puede jugar, comprobar estadísticas personales y visualizar el ranking.

Cuando juega, tiene 6 intentos, puede fallar, si falla, vuelve a jugar (que sería escribir otra palabra), así hasta que gane porque la palabra haya sido acertada o porque se han acabado todos los intentos y pierda.



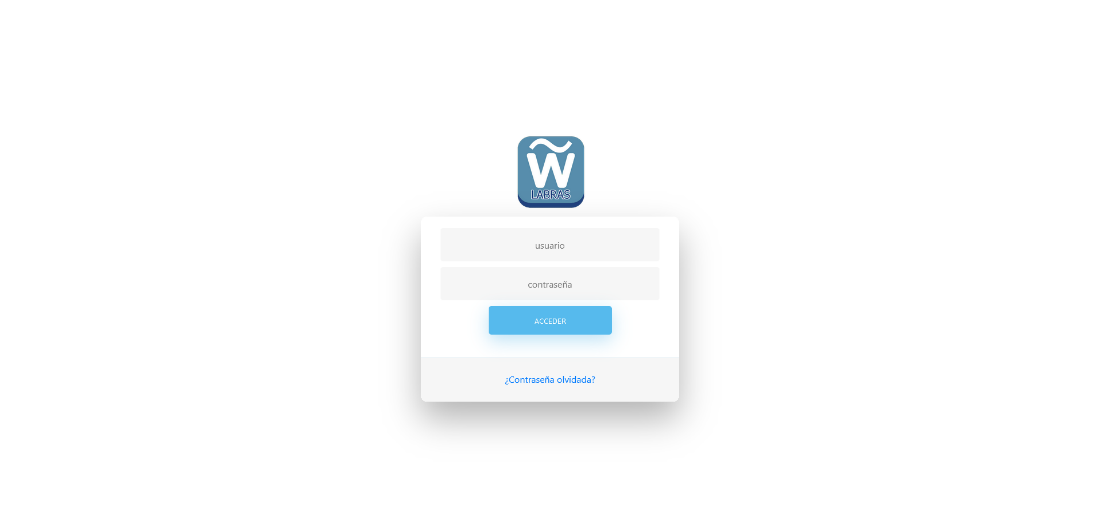
## Interfaz

### Características generales

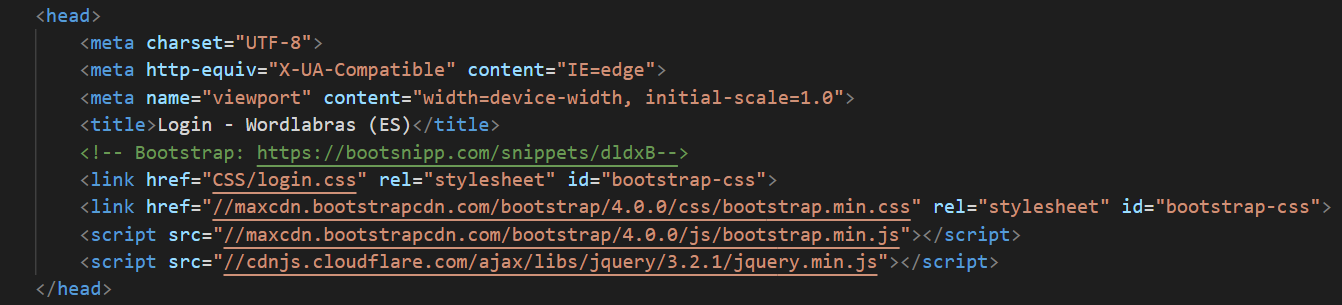
#### Descripción de la interfaz

#### Características generales del diseño y la navegación

Pantalla del login creada con ayuda de bootstrap, cuenta con efectos ‘fade-in’ y ‘face-out’ realizados con CSS

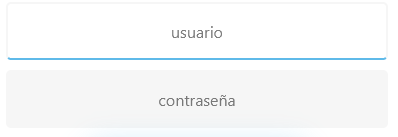


Importe del CSS y el boostrap/jquery utilizado para los efectos y visualización del login.

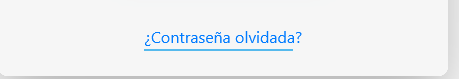


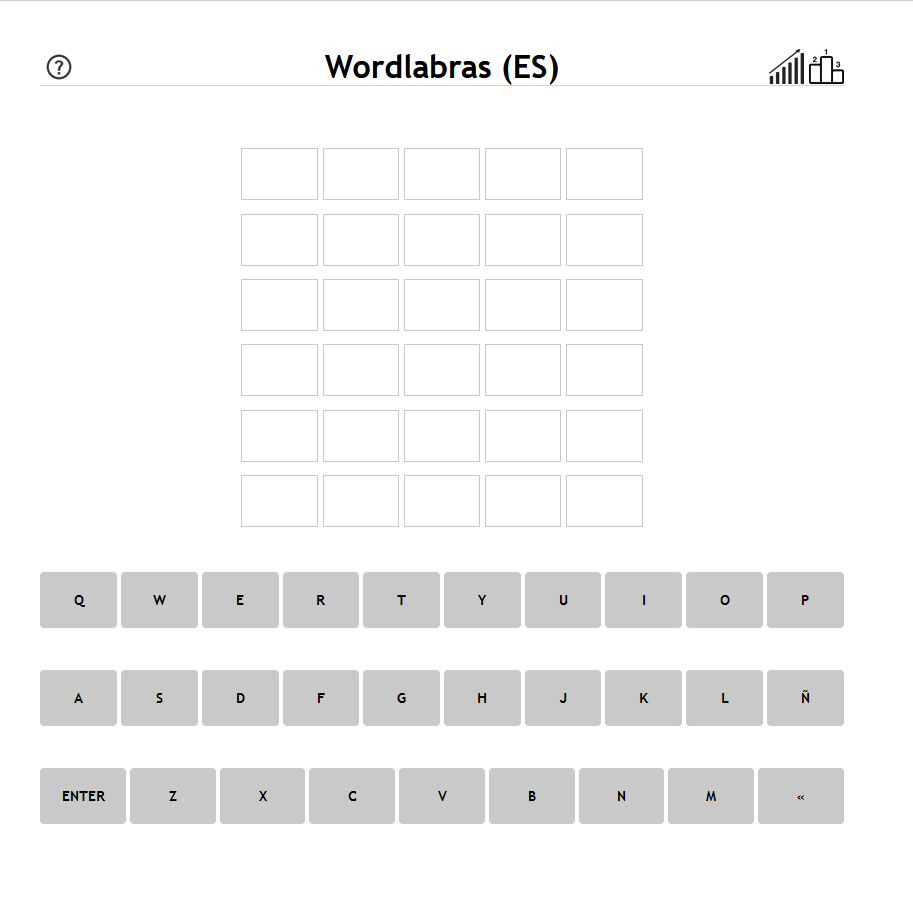
Cuenta con efectos de transiciones in/out para todo el contenido del login además de efectos “hover” para cuando el ratón sea posicionado sin clickar en los campos, véase en “acceder” que oscurecerá el color azul al tener el ratón encima del botón. Los inputs del formulario también contienen efectos hover y por click, haciendo que obtengan una línea de color azul y cambiar un poco el color de fondo y sombreados.

Ejemplo:

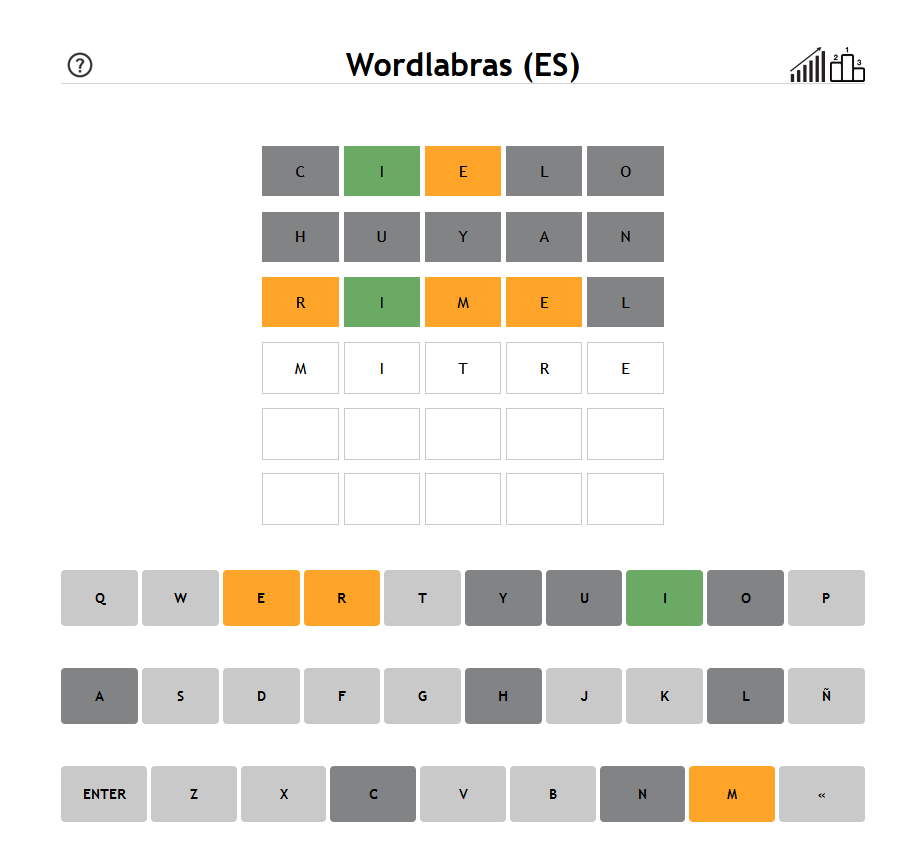


Ocurre exactamente lo mismo con “contraseña olvidada”, cuenta con un subrayado azul con efecto (no aparece sin más, si no que se va generando de izquierda a derecha al tener el ratón posado en el botón y cambia el color a azul por ser un hipervínculo).

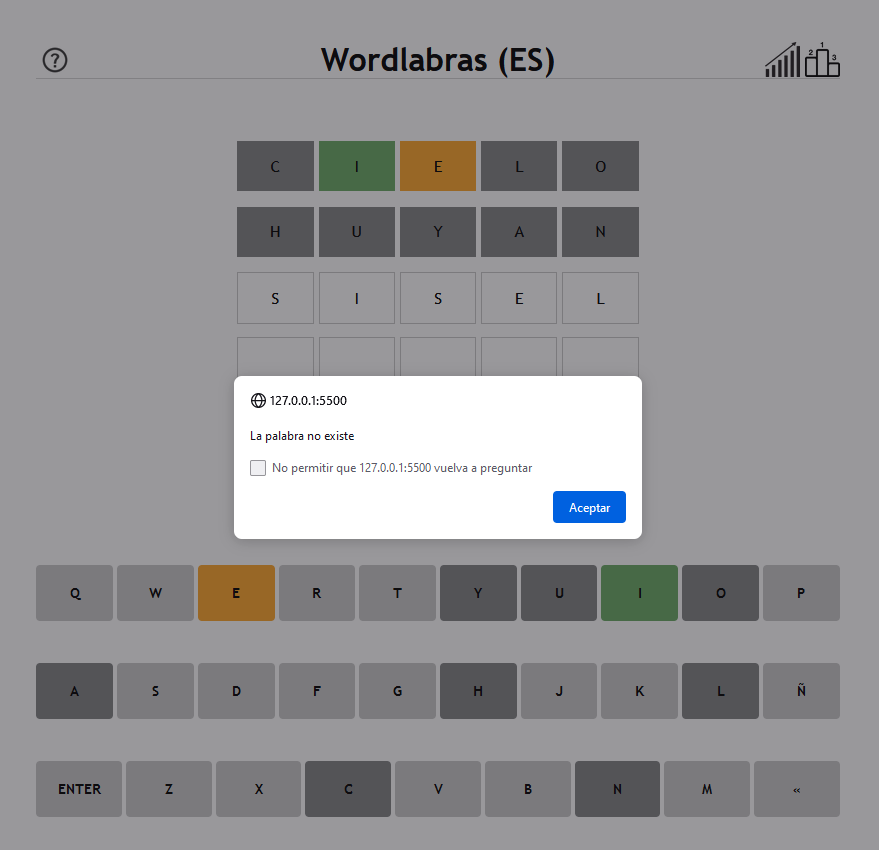


En la interfaz del juego, podemos ver que tiene una visualización como la siguiente: 

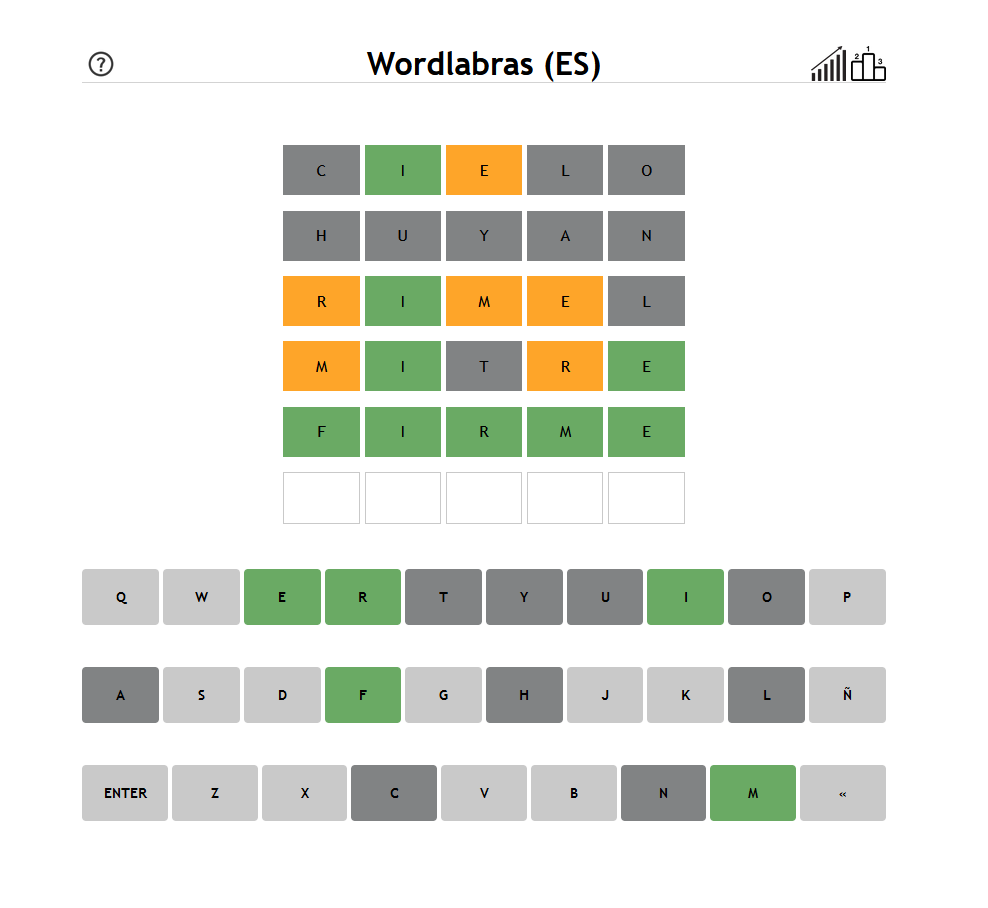
La aplicación al introducir una palabra que existe, mostrará en blanco los intentos y dentro las letras escritas si aún no introdujo la palabra pulsando ENTER. Una vez pulse ENTER se pondrán en color gris oscuro los cuadrados de las letras que no existen en la palabra a adivinar, en naranja las letras que existen en la palabra a adivinar pero que no están en la posición correcta y de color verde cuando la letra introducida y su posición coinciden con la de la palabra a adivinar.



Cuando escribe una palabra que no existe, la aplicación mostrará una alerta advirtiendo que la palabra escrita no existe



Una vez haya ganado, la palabra a adivinar se pondrá en verde y mostrará una alerta de que has ganado además de la opción de visualizar las estadístidas o el ranking



Al visualizar las estadísticas, nos saldrá una tabla con todas nuestras partidas, el número de victorias y el número de intentos gastados

En la página del ranking se mostrará una tabla con todos los usuarios, el que más victorias tenga estará primero, es decir, el número de victorias es el factor determinante para el sistema de puntuaje, en caso de empate, se contarán el número de intentos, y en caso de empate por ambas cosas, será por orden alfabético

#### Lenguajes empleados y validación

Los lenguajes empleados han sido:

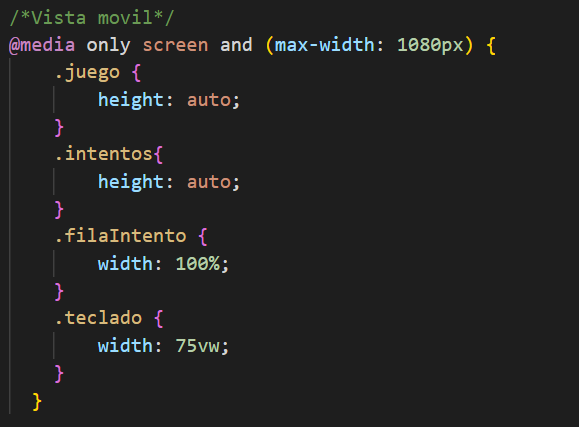
* HTML
  + Se ha utilizado para la visualización y creación de nodos base/iniciales para el juego. También es el apartado principal de las páginas web.
* CSS
  + Utilizado bastante en todo el proyecto para su correcta visualización y su estilo. Hace que todo se vea tan bien como se espera.
  + Utilizando Bootstrap en el login
* Javascript
  + Se ha utilizado en todo el proyecto para cualquier tipo de interacción, desde dejar el rato encima de la barra para introducir el usuario, como la creación del teclado virtual así como de las cajas de los intentos de palabra así como recoger las teclas pulsadas correctas y el envío de datos a la base de datos.
  + Node.js
  + Axios.js
* SQL
  + Utilizado para la base de datos relacional y para la inyección de datos por medio de JavaScript gracias al uso de algunas librerías que posee (Node.js, Express.js, etc...)

Las validaciones que tendrán serán para los usuarios, palabras y letras introducidas en el juego, registros de usuario (no-realizado), y los datos metidos y sacados por pantalla de la base de datos (estadísticas y ranking).

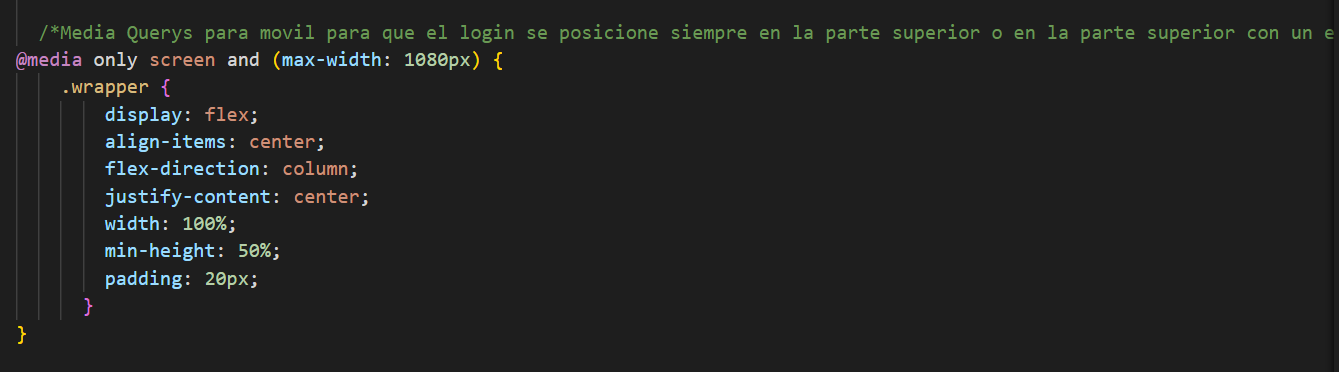
### Adaptación a dispositivos móviles

La adaptación se ha realizado a través del uso de “mediaquerys” en el código CSS.

#### Mediaquery para la interfaz del juego

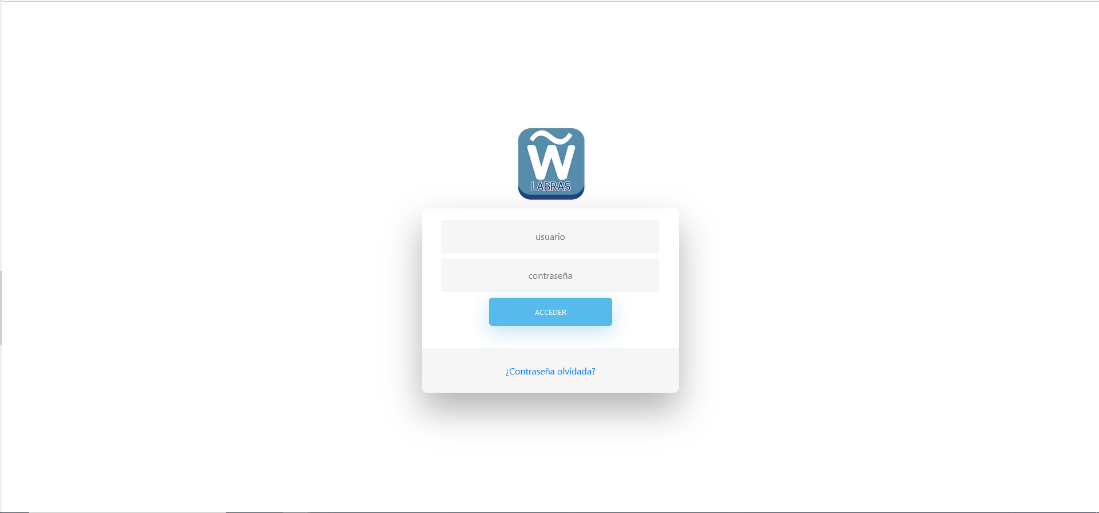


#### MediaQuery para la interfazl del Login

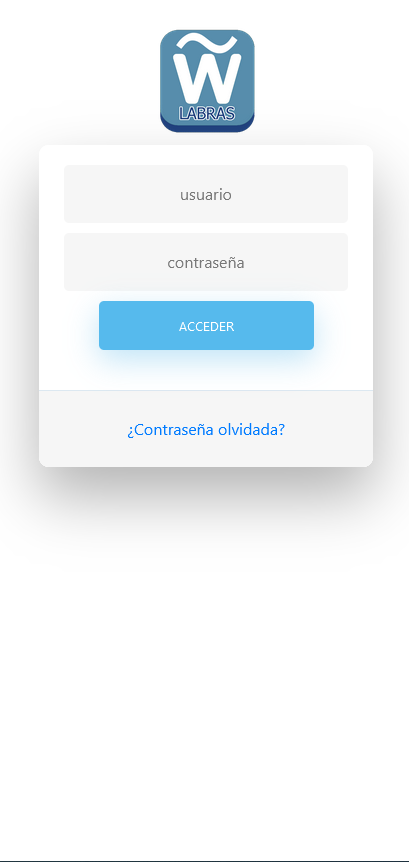


Las mediaquerys para las estadísticas y para el ranking serán más o menos similares, y se visualizarán similarmente a la interfaz del juego ya que será una página donde se visualice una tabla en ambos casos.

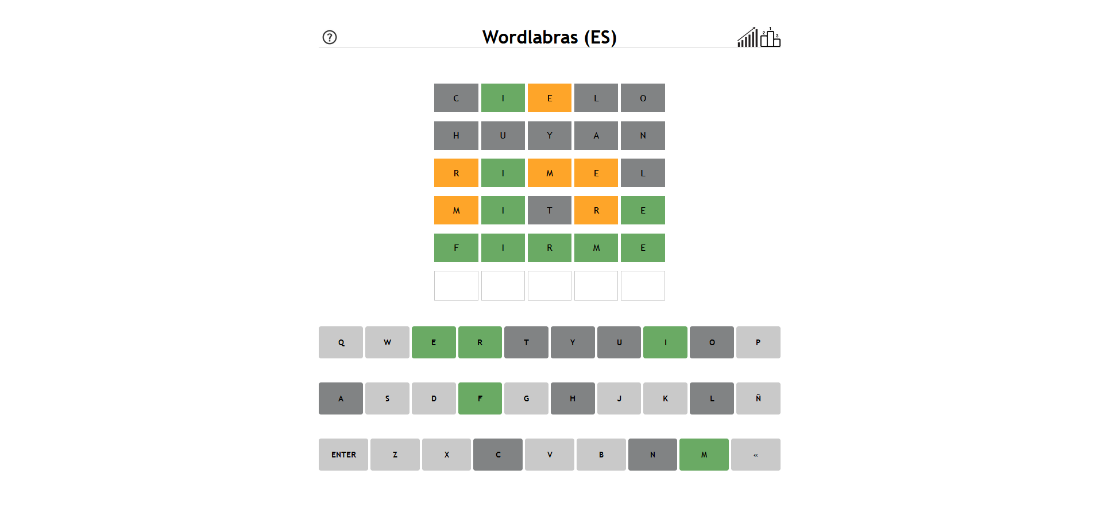
#### Vista ordenador del login



#### Vista móvil del login

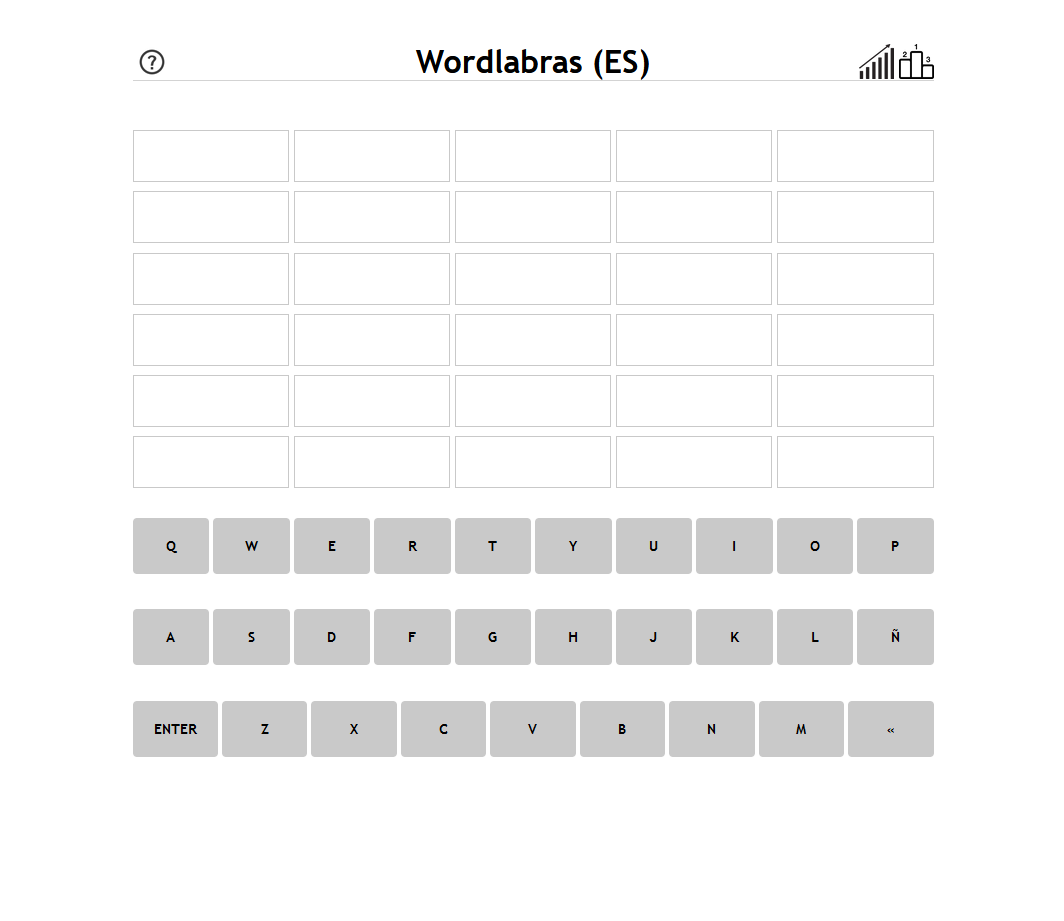


#### Vista ordenador del juego

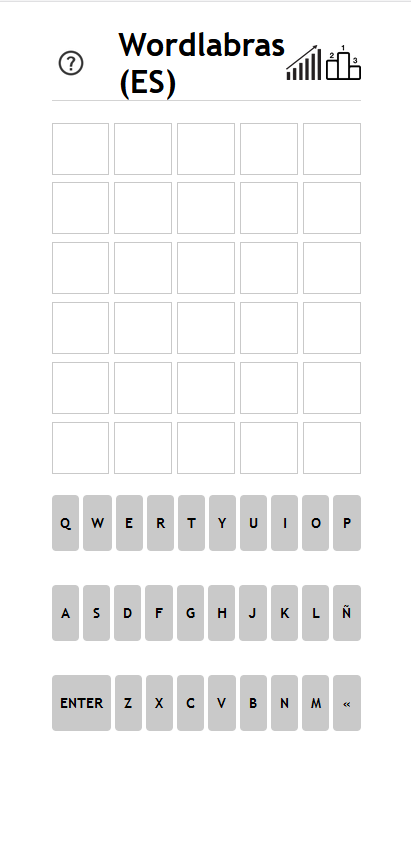


#### Vista móvil/tablet del juego en distintas resoluciones

##### Vista tablet del juego



##### Vista móvil del juego



### Usabilidad/accesibilidad

#### Usabilidad

La usabilidad ha sido implementada para que el juego pueda jugarse tanto desde teclado como desde ratón. Puedes utilizar sólo el teclado para teclear las palabras y meter la palabra o borrar letras como el ratón para usar el teclado virtual. Para navegar por las ventanas es cierto que se deberá hacer uso exclusivo del ratón.

#### Accesibilidad

La accesibilidad será global ya que estará disponible para ser visualizado por cualquier dispositivo genérico de una forma correcta, ya sea un ordenador (nativo), como un móvil (adaptado) o una tablet (adaptado). No contará con soporte interno hacia otro tipos de dispositivo al menos nativos, pero eso no quiere decir que desde los propios distintos dispositivos no se pueda adaptar para la correcta accesibilidad y visualización, ya que estará colgado de forma púbica en una página web y en principio la accesibilidad será global.

# Autoevaluación y conclusiones

## Valoración del trabajo y dificultades encontradas

Mi trabajo actual lo valoro de tal manera que la mayor parte del trabajo está en el javascript y en el CSS así como en la BBDD, las conexiones y el enlace entre el juego y la base de datos por medio de javascript. Haciendo uso de Node.js y Axios ha sido posible. Ésta es la parte innovadora del proyecto. La parte del login si bien es cierta que es “floja” debido a que ha sido realizada prácticamente con boostrap, creo que no era una parte importante del proyecto a visualizar y el tiempo escaseaba. No creo tampoco que sea una parte complicada la realización de un HTML/CSS para una página de login por lo que no veo que tenga mucho peso. He dedicado bastante tiempo las dos últimas semanas de entrega del proyecto a la realización del mismo para que pueda ser presentable. Creo que aún le quedan muchas cosas que me hubiese gustado implementar y muchas otras por mejorar, creo que este proyecto podría haber sido mucho más vistoso y bonito con más tiempo dedicado. Por ejemplo; la implementación de que el usuario decida con cuántas letras tiene la palabra que quiere adivinar y que se auto genere el juego en base a esa opción que escoja. Tenía preparado el TXT y la API de la RAE para hacerlo de esa manera, pero me vi obligado a quitarla por la falta de tiempo y el tiempo que suponía adaptar esa idea con el código a medio realizar. Veía otras partes más importantes como es la base de datos y decidí centrarme mejor en eso. También me hubiese gustado implementar la API que encontré para meter el diccionario de la RAE pero sólo vi opciones a través de Python y no tuve tiempo para pensar en cómo poder implementarlo desde Javascript u otra alternativa. Creo que pude hacer mucho más si hubiese dedicado más tiempo, pero simplemente no se dio porque estaba con las prácticas y a otras cosas personales de interés.

Las dificultades encontradas han sido:

* Por supuesto el tiempo invertido. Creo que no he invertido lo suficiente para lo que me gustaría y quizás incluso después de toda la entrega, mejore el proyecto e implemente poco a poco lo que se me ocurrió por practicar la codificación y programación.
* Entendimiento de la memoria, hay partes que ha sido difícil de hacer y entender, hay algún diagrama que no recuerdo haberlo visto en clase (diagrama secuencial y diagrama de objetos).
* Aprendizaje de node.js y axios (que utiliza node.js). No fue como tal dificultad, pero fue algo nuevo que aprender
* La lógica ha supuesto algo de reto, sobretodo para que funcione correctamente un poco a base de ver posibles errores e ir corrigiéndolos.
* CSS vistoso, costó recordar un poco las ocpiones que tenía

## Valoración de la herramienta o aplicación desarrollada

La herramienta en sí me parece muy buena y estable, sobretodo sin el uso de la base de datos. Tiene varias mejoras por implementar para su mejor diseño y funcionalidad con la base de datos, como pueda ser la seguridad y la estructura. El código me parece limpio, aunque se ha realizado casi desde 0 sin hacer uso de frameworks que hubiesen ahorrado tiempo. En sí la idea es muy divertida tanto de realizar como de probar, la herramienta como tal tiene ya un predecesor que ha sido valorado muy positivamente y del que se han diversificado diferentes versiones, esta aplicación podría decirse que no es más que otra diversificación. El implementar un ranking creo que es una mejora positiva para la aplicación.

## Conclusiones finales

### Del diseño

El diseño tenía pensado diferenciarlo un poco más respecto al de Wordle pero no me ha dado tiempo a centrarme demasiado en ese aspecto y al tener la referencia de Wordle pues intenté que quedase bien basándome en esa referencia. Los colores escogidos además para el acierto o no de las letras en la palabra a adivinar me parecían un acierto y una pena cambiarlos (gris, verde y naranja).

El diseño del Login está hecho con bootstrap y la verdad es que no tengo mucho que decir sobre el diseño, creo que está perfecto, le faltaría por implementar el registro y con ello los cambios necesarios para una mayor seguridad (añadir la página correspondiente de registro y añadir el apartado de email para el usuario) pero en general está bien.

### De la aplicación

La aplicación me gusta mucho, me hubiese gustado haber filtrado un poco más las palabras escogidas para jugar, pero al ser demasiadas porque están cogidas de la RAE y ya toqué bastante pues no quise tocar más ni dedicarle más tiempo, pero hay verbos en diferentes tiempos como “pasan” y “pasar” que no me gustaría que estuvieran.

### Del tiempo empleado

El tiempo que he empleado ha sido aproximadamente entre 64h-120h. No estimo muy bien cuántas horas pudieron ser ya que no hice un seguimiento y ha sido cuanto menos irregular, pero podría decirse que fueron más o menos rondando esas horas.

### De los conocimiendos adquiridos

Los conocimientos adquiridos han sido sobre todo node.js, axios.js y la manera de conectar con la base de datos desde JavaScript sumado a poder hacer sentencias a la base de datos desde Javascript. También me ha venido muy bien para recordar lo aprendido durante el curso y lo agradezco bastante ya que creo que ha sido bastante útil. También me ha motivado a seguir programando y haciendo proyectos personales para mi futuro desarrollo así como porque me gusta.

# Bibliografía

1. HTML: <https://developer.mozilla.org/es/docs/Web/HTML>
2. CSS: <https://developer.mozilla.org/es/docs/Web/CSS>
3. JavaScript: <https://developer.mozilla.org/es/docs/Web/JavaScript>
4. Node.js: <https://es.wikipedia.org/wiki/Node.js>
5. Descarga: <https://nodejs.org/en/download/>
6. Axios: <https://axios-http.com/es/docs/intro>
7. SQL: <https://es.wikipedia.org/wiki/SQL>
8. MySQL: [Enlace](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjon97U5uz7AhWux4UKHf5ICIAQmhN6BAhhEAI&url=https%3A%2F%2Fes.wikipedia.org%2Fwiki%2FMySQL&usg=AOvVaw2AcYhwh8LjZo-SUx7BRaVP)
9. VisualStudio Code: <https://code.visualstudio.com/>
10. Enlace: <https://www.apachefriends.org/es/index.html>

1. HTML: <https://developer.mozilla.org/es/docs/Web/HTML> [↑](#footnote-ref-1)
2. CSS: <https://developer.mozilla.org/es/docs/Web/CSS> [↑](#footnote-ref-2)
3. JavaScript: <https://developer.mozilla.org/es/docs/Web/JavaScript> [↑](#footnote-ref-3)
4. Node.js: <https://es.wikipedia.org/wiki/Node.js>

   Descarga: <https://nodejs.org/en/download/> [↑](#footnote-ref-4)
5. Axios: <https://axios-http.com/es/docs/intro> [↑](#footnote-ref-5)
6. SQL: <https://es.wikipedia.org/wiki/SQL> [↑](#footnote-ref-6)
7. MySQL: [Enlace](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwjon97U5uz7AhWux4UKHf5ICIAQmhN6BAhhEAI&url=https%3A%2F%2Fes.wikipedia.org%2Fwiki%2FMySQL&usg=AOvVaw2AcYhwh8LjZo-SUx7BRaVP) [↑](#footnote-ref-7)
8. VisualStudio Code: <https://code.visualstudio.com/> [↑](#footnote-ref-8)
9. Enlace: <https://www.apachefriends.org/es/index.html> [↑](#footnote-ref-9)