Características y funcionalidades del TFG: Wordle+.

Alumno: David Correa Rodríguez

Tutor: Juan Manuel Fernández Luna

**Breve descripción:** el TFG tratará de desarrollar el típico juego de palabras Wordle, pero añadiéndole aspectos multijugador y competitivos para que tenga más entidad de juego completo. Se podrán gestionar torneos entre jugadores y retos 1vs1, gestionando un ranking de jugadores e históricos de partidas por jugador. Los jugadores se dividirán por rangos, en relación con la experiencia de juego de cada jugador (Wordles, torneos y retos completados). Las partidas podrán ser más personalizables, como por ejemplo poder elegir la longitud de la palabra en cuestión.

Por tanto, podemos dividir las características y funcionalidades de este proyecto en 3 niveles, en donde el primer nivel corresponde a las funcionalidades obligatorias, el segundo nivel a las funcionalidades muy recomendadas, y el tercer nivel corresponde a aquellas funcionalidades extra que se añadirán si el tiempo y las condiciones lo permiten:

**Nivel 1:**

* Registro de jugadores y de gestores de eventos.
* Inicio de sesión.
* Poder jugar la partida clásica de Wordle.
* Poder configurar la partida en cuestión (longitud de palabra).
* Los jugadores tendrán atributos asociados: Nivel de Experiencia, Rango (asociado al nivel de experiencia: leyenda, maestro, veterano, principiante, iniciado), Wordles completados, retos ganados, torneos ganados
* Gestión de retos 1vs1 entre usuarios amigos.
* Gestión de torneos: se podrán crear salas de torneos de distintos tamaños en donde los usuarios podrán unirse; se podrán organizar torneos por los gestores de eventos quienes seleccionarán los participantes de los torneos.

**Nivel 2:**

* Gestionar un ranking con los mejores jugadores, que se divida por filtros (partidas ganadas, retos ganados, torneos ganados, nivel de experiencia)
* Gestionar un histórico de Wordles completados y retos 1vs1 por jugador
* Implementación de un bot que juegue automáticamente a las partidas, con la posibilidad de añadirlo a los torneos y comprobar su comportamiento.
* Los torneos se dividirán según los rangos de los jugadores para que la experiencia de juego sea justa y los enfrentamientos sean equilibrados.
* Los torneos, además de dividirse por rango, también se dividirán por longitud de palabras (Torneo de maestros de 7 letras, torneo de iniciados de 4 letras, etc)

**Nivel 3:**

* Mejoras de rendimiento
* Mejoras visuales y de diseño
* Implementación de multilenguaje (Español/Inglés)
* Implementar un modo oscuro (dark mode)
* Desarrollar, además de una app web, una aplicación móvil, tanto para Android como para iOS, y poder jugar al juego mediante un smartphone.

Me gustaría destacar que la única funcionalidad existente en el juego Wordle es la segunda funcionalidad del primer nivel (Poder jugar la partida clásica de Wordle) y la penúltima del nivel 3 (Implementar un modo oscuro)

Metodología e historias de usuario del TFG: Wordle+.

Alumno: David Correa Rodríguez

Tutor: Juan Manuel Fernández Luna

***Metodología***

A comienzos de la década de los 90 muchos desarrolladores software se percataron de que los ciclos de producción y los métodos que ofrecían las metodologías clásicas no estaban dando resultados satisfactorios. En un entorno lleno de incertidumbre, tanto empresarial como económico, muchos proyectos software se veían cancelados antes de lanzarse al mercado.

Este fue el origen de la creación de las metodologías ágiles, que florecieron a partir de la década de los 2000 con la creación del Manifiesto Ágil[[1]](#endnote-1). Los valores de la metodología ágil sostienen que:

* ***Las personas y las interacciones*** *antes que los procesos y las herramientas*
* ***El software en funcionamiento*** *antes que la documentación exhaustiva*
* ***La colaboración con el cliente*** *antes que la negociación contractual*
* ***La respuesta ante el cambio*** *antes que el apego a un plan*

En donde los puntos de la izquierda son más importantes que los de la derecha. Por tanto, lo que permite este tipo de metodologías es evitar desarrollar sistemas software en secuencia por fases y utilizar un proceso de desarrollo simultáneo y constante.

Con la filosofía asentada, se crearon diversos marcos ágiles enfocados al desarrollo del software, como Scrum[[2]](#endnote-2), Kanban[[3]](#endnote-3) o Programación Extrema (XP), que actualmente son piezas fundamentales del [DevOps o](https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-devops/) la integración continua/implementación continua [CI/CD](https://unity.com/es/solutions/what-is-ci-cd)[[4]](#endnote-4)[.](https://unity.com/es/solutions/what-is-ci-cd)

Scrum2 es uno de los marcos de trabajo más utilizados en la actualidad, que permite el trabajo colaborativo entre equipos de desarrollo de software, aunque sus valores y principios se pueden aplicar a diversos tipos de trabajo colaborativo.

Scrum2 es un marco de trabajo heurístico, que se basa en el aprendizaje y continuo y en la readaptación de los factores fluctuantes del equipo. Supone que el equipo no sabe las variables iniciales de un proyecto y que evolucionará con él. Scrum2 hace uso de “artefactos”, que son herramientas para solucionar un problema. Esta metodología utiliza tres de ellos:

* **Product backlog:** es la pila o la lista inicial del trabajo que tiene que realizar el propietario del producto, en donde se incluyen requisitos, funcionalidades y mejoras del producto. Esta lista no es absoluta, si no que se revisa y se cambia por el propietario del producto, ante las fluctuaciones del mercado y los requisitos del propietario.
* **Sprint backlog:** se trata de la lista de elementos, historias de usuario, seleccionada por el equipo de trabajo para su implementación en el sprint actual. Un sprint es un pequeño periodo en el cual el equipo de desarrollo implementa y distribuye un incremento del producto (una versión del producto). Los sprints suelen tener una duración de 2 semanas, aunque puede cambiar según el equipo o el trabajo a realizar. Esta lista de elementos del sprint es flexible y puede evolucionar con el sprint.
* **Incremento:** es el producto final utilizable de un sprint, es decir, el resultado del sprint. También se le suele llamar Epic. La definición tampoco es rígida y depende del contexto y el entorno, puede ser que un incremento no suponga una versión del producto final, si no una parte de él.

Una característica fundamental en la metodología Scrum2 es el ***scrum diario***o reunión rápida, de unos 10-15 minutos de duración, en donde el equipo hace hincapié en lo que se debe se hacer ese día, además de comentar qué se hizo el día anterior. También se utiliza para expresar inquietudes e impedimentos del sprint actual.

Al finalizar un sprint se suele hacer una ***revisión del sprint*** en donde el equipo demuestra e inspecciona el incremento, resultado de haber desarrollado los elementos del backlog del sprint. Esta revisión también es útil para que el propietario vea el incremento y decida si lanzarlo al mercado o no.

Otro aspecto importante que destacar es la ***retrospectiva del sprint*** que sirve para que el equipo documente y analice los aspectos del sprint que han funcionado y los que no. En esta etapa el equipo puede analizar lo que salió mal y evolucionar para un mejor desarrollo software.

En esta metodología existen tres figuras importantes resaltables:

* El propietario del producto o Product Owner (PO): son los que más conocen el producto y lo proponen. Se centran en los aspectos empresariales, de los clientes y del mercado. Básicamente define el backlog del producto, está en contacto constante con el equipo de desarrollo y decide cuándo lanzar el producto.
* El experto en scrum o Scrum Master (SM): son los que proporcionan formación al equipo y a los propietarios del producto para refinar su práctica. Básicamente planifica los recursos necesarios para realizar los sprints, las reuniones, las revisiones y las retrospectivas.
* El equipo del desarrollo: son los que realizan el trabajo y desarrollan el producto en varios sprints. Los miembros se reorganizan y aprenden entre ellos para tener una actitud positiva colectiva.

Por tanto, para este proyecto, se utilizará una metodología Scrum2 adaptado a 1 sola persona, es decir, tanto el dueño del producto, el Scrum Master como el equipo de desarrollo lo conformaré yo mismo, tomando los diferentes roles y adoptando diferentes perspectivas según el rol.

Se seguirán las etapas de planificación (product backlog), ejecución (sprints, pila de sprint, probablemente cortos, de 2 semanas) y control (revisiones con el profesor y retrospectiva). El profesor podría tomar de forma provisional el rol de cliente y juzgar de forma objetiva los resultados obtenidos.

Esta cuestión ha sido consultada a la profesora de la asignatura Metodologías de Desarrollo Ágil (mlra@ugr.es).

Al ser un proyecto largo, me beneficiaré de las diversas versiones que permite la metodología scrum, para así obtener un producto más refinado y de calidad. También haré toma de contacto con esta metodología, muy utilizada actualmente en el mercado y en multitud de empresas.

***Historias de usuario.***

**DEFINICIONES Y ACLARACIONES PREVIAS**

**ROLES EN EL SISTEMA: JUGADOR, GESTOR DE EVENTOS (GE) Y USUARIO**

**USUARIO: TANTO JUGADOR COMO GESTOR DE EVENTOS**

**DATOS PERSONALES: NOMBRE, APELLIDOS, NICKNAME, ETC**

# PERFIL DE JUGADOR: PARTIDAS GANADAS, PARTIDAS COMPLETADAS, RANGO, RETOS GANADOS, ETC

Los puntos de historia (PH) son una unidad de medida utilizada en proyectos ágiles para estimar la carga global de una historia de usuario. En este caso, se ha utilizado como referencia la escala de Fibonacci.

|  |  |  |  |
| --- | --- | --- | --- |
| **Ident.** | **Título** | **Estimación** | **Prioridad** |
| **HU.1** | Como jugador quiero poder registrarme | 2 | 1 |
| **HU.2** | Como gestor de eventos quiero poder registrarme | 2 | 1 |
| **HU.3** | Como usuario (jugador y gestor) quiero poder iniciar sesión | 2 | 1 |
| **HU.4** | Como usuario quiero poder cerrar sesión | 1 | 1 |
| **HU.5** | Como jugador quiero poder echar un Wordle clásico | 8 | 1 |
| **HU.6** | Como jugador quiero poder  seleccionar la longitud de la palabra de la partida clásica | 2 | 2 |
| **HU.7** | Como gestor de eventos quiero crear una sala abierta de un torneo | 5 | 2 |
| **HU.8** | Como GE quiero poder seleccionar el tamaño del torneo (número de  participantes) | 2 | 2 |
| **HU.9** | Como GE quiero poder seleccionar la longitud de la palabra del torneo | 2 | 2 |
| **HU.10** | Como GE quiero poder crear un torneo y preseleccionar los participantes | 3 | 2 |
| **HU.11** | Como jugador quiero poder jugar una partida 1vs1 con un amigo | 5 | 2 |
| **HU.12** | Como jugador quiero poder seleccionar la longitud de la palabra del torneo (tipo de torneo) | 3 | 2 |
| **HU.13** | Como jugador quiero poder unirme a una sala abierta de un torneo | 2 | 2 |
| **HU.14** | Como usuario quiero poder modificar mis datos personales | 2 | 3 |
| **HU.15** | Como jugador quiero poder ver el ranking | 5 | 3 |
| **HU.16** | Como jugador quiero poder seleccionar el filtro del ranking | 2 | 3 |
| **HU.17** | Como jugador quiero poder ver mi perfil de jugador | 2 | 3 |
| **HU.18** | Como usuario quiero poder cambiar el idioma de la plataforma | 2 | 4 |
| **HU.19** | Como jugador quiero poder ver mi lista de amigos | 3 | 2 |
| **HU.20** | Como jugador quiero poder buscar a otro jugador por su nombre y mandarle  una solicitud de amistad | 3 | 2 |
| **HU.21** | Como jugador quiero poder aceptar una solicitud de amistad | 2 | 2 |
| **HU.22** | Como jugador quiero poder eliminar un jugador de mi lista de amigos | 1 | 4 |
| **HU.23** | Como jugador quiero poder ver el perfil de jugador de un amigo | 2 | 4 |
| **HU.24** | Como usuario quiero poder cambiar el modo de visualización de la plataforma (modo predeterminado / modo oscuro) | 3 | 5 |
| **HU.25** | Como jugador quiero poder ver mis notificaciones a través del buzón | 2 | 2 |

|  |  |
| --- | --- |
| **Identificador:** HU1 | Registro jugador |
| **Descripción:** Como jugador quiero poder registrarme | |
| **Estimación** 2  **Prioridad** 1  **Entrega** | |
| **Pruebas de aceptación:**   * Introducir un nickname ya existente en el sistema y comprobar que se indica un error * Introducir los campos vacíos y comprobar que se indica un error * Introducir caracteres inválidos y comprobar que se indica un error * Introducir todos los datos correctos y comprobar que se ha registrado un nuevo jugador en la base de datos | |
| **Observaciones:** | |

|  |  |
| --- | --- |
| **Identificador:** HU2 | Registro gestor de eventos |
| **Descripción:** Como gestor de eventos quiero poder registrarme | |
| **Estimación** 2  **Prioridad** 1  **Entrega** | |
| **Pruebas de aceptación:**   * Introducir un nickname ya existente en el sistema y comprobar que se indica un error * Introducir un código de autorización incorrecto y comprobar que se indica un error. * Introducir los campos vacíos y comprobar que se indica un error * Introducir caracteres inválidos y comprobar que se indica un error * Introducir todos los datos correctos y comprobar que se ha registrado un nuevo jugador en la base de datos | |
| **Observaciones:** | |

|  |  |
| --- | --- |
| **Identificador:** HU3 | Iniciar sesión |
| **Descripción**: Como usuario (jugador y gestor) quiero poder iniciar sesión | |
| **Estimación** 2  **Prioridad** 1  **Entrega** | |
| **Pruebas de aceptación:**   * Introducir un nickname no registrado en el sistema y comprobar que se indica un error * Introducir una contraseña incorrecta y comprobar que se indica un error * Introducir los campos vacíos y comprobar que se indica un error * Introducir caracteres inválidos y comprobar que se indica un error * Introducir todos los datos correctos y comprobar que el usuario ha accedido a las funcionalidades de la aplicación / se encuentra en la lista de usuarios   conectados en el sistema | |
| **Observaciones:** | |

|  |  |  |
| --- | --- | --- |
| **Identificador:** HU4 | Cerrar sesión | |
| **Descripción**: Como usuario quiero poder cerrar sesión | | |
| **Estimación** 1 | **Prioridad** 1 | **Entrega** |
| **Pruebas de aceptación:**   * Comprobar que el usuario no se encuentra en la lista de usuarios conectados en el sistema | | |
| **Observaciones:** | | |

|  |  |
| --- | --- |
| **Identificador:** HU5 | Wordle clásico |
| **Descripción**: Como jugador quiero poder jugar un Wordle clásico | |
| **Estimación** 8  **Prioridad** 1  **Entrega** | |
| **Pruebas de aceptación:**   * Comprobar que el jugador introduce un intento de palabra y se indica la posición de cada letra (contiene, no contiene, y acierto) * Comprobar que el jugador agota todos los intentos y se indica un mensaje de que ha perdido la partida * Comprobar que el jugador adivina la palabra y se indica un mensaje de que ha ganado la partida | |
| **Observaciones:** | |

|  |  |  |
| --- | --- | --- |
| **Identificador:** HU6 | Longitud palabra Wordle clásico | |
| **Descripción**: Como jugador quiero poder seleccionar la longitud de la palabra de la partida clásica | | |
| **Estimación** 2 | **Prioridad** 2 | **Entrega** |
| **Pruebas de aceptación:**   * Comprobar que el jugador selecciona la nueva longitud de palabra y se aplican los cambios en la aplicación * Comprobar que el jugador no puede escribir en el selector de longitud de palabra * Comprobar que el jugador, al escribir un intento, su longitud corresponde con el introducido en el selector de longitud | | |
| **Observaciones:** | | |

|  |  |  |
| --- | --- | --- |
| **Identificador:** HU7 | Crear sala torneo | |
| **Descripción**: Como gestor de eventos quiero crear una sala abierta de un torneo | | |
| **Estimación** 5 | **Prioridad** 2 | **Entrega** |
| **Pruebas de aceptación:**   * Introducir un valor no válido de número de participantes de la sala y comprobar que se muestra un mensaje de error * Introducir un tamaño de palabra no válido y comprobar que se muestra un mensaje de error * Introducir los campos vacíos y comprobar que se muestra un mensaje de * error * Introducir todos los campos correctamente y que se crea un nuevo torneo en la lista de torneos disponibles en el sistema | | |
| **Observaciones:** | | |

|  |  |  |
| --- | --- | --- |
| **Identificador:** HU8 | Tamaño torneo | |
| **Descripción**: Como GE quiero poder seleccionar el tamaño del torneo (número de participantes) | | |
| **Estimación** 2 | **Prioridad** 2 | **Entrega** |
| **Pruebas de aceptación:**  • Comprobar que el tamaño indicado coincide con el tamaño de la sala recién creada | | |
| **Observaciones:** | | |

|  |  |  |
| --- | --- | --- |
| **Identificador:** HU9 | Tamaño palabra torneo | |
| **Descripción**: Como GE quiero poder seleccionar la longitud de la palabra del torneo | | |
| **Estimación** 2 | **Prioridad** 2 | **Entrega** |
| **Pruebas de aceptación:**  • Comprobar que el tamaño de palabra indicado coincide con el tamaño de palabra de la sala recién creada | | |
| **Observaciones:** | | |

|  |  |  |
| --- | --- | --- |
| **Identificador:** HU10 | Crear torneo fijo | |
| **Descripción**: Como GE quiero poder crear un torneo y preseleccionar los participantes | | |
| **Estimación** 3 | **Prioridad** 2 | **Entrega** |
| **Pruebas de aceptación:**   * Comprobar que se introducen usuarios registrados en el sistema * Comprobar que en número de usuarios preseleccionados no supera el tamaño máximo del torneo * Introducir un valor no válido de número de participantes de la sala y comprobar que se muestra un mensaje de error * Introducir un tamaño de palabra no válido y comprobar que se muestra un mensaje de error * Introducir los campos vacíos y comprobar que se muestra un mensaje de * error * Introducir todos los campos correctamente y que se crea un nuevo torneo en la lista de torneos disponibles en el sistema * Comprobar que al jugador invitado le ha llegado la invitación | | |
| **Observaciones:** | | |

|  |  |  |
| --- | --- | --- |
| **Identificador:** HU11 | Crear partida 1vs1 | |
| **Descripción**: Como jugador quiero poder jugar una partida 1vs1 con un amigo | | |
| **Estimación** 5 | **Prioridad** 2 | **Entrega** |
| **Pruebas de aceptación:**   * Comprobar que la palabra asignada es la misma para ambos jugadores en la partida * Comprobar que, al terminar, la partida se ha introducido en el histórico de partidas * Comprobar que al jugador invitado le ha llegado la invitación de la partida | | |
| **Observaciones:** | | |

|  |  |
| --- | --- |
| **Identificador:** HU12 | Seleccionar longitud palabra torneo jugador |
| **Descripción**: Como jugador quiero poder seleccionar la longitud de la palabra del torneo (tipo de torneo) | |
| **Estimación** 3  **Prioridad** 2  **Entrega** | |
| **Pruebas de aceptación:**   * Comprobar que al usuario se le muestran las salas abiertas de torneos que tienen la misma longitud de palabra que la seleccionada | |
| **Observaciones:** | |

|  |  |  |
| --- | --- | --- |
| **Identificador:** HU13 | Unirse a sala | |
| **Descripción:** Como jugador quiero poder unirme a una sala abierta de un torneo | | |
| **Estimación** 2 | **Prioridad** 2 | **Entrega** |
| **Pruebas de aceptación:**   * Comprobar que el jugador aparece en la lista de usuarios inscritos a la sala * Comprobar que el jugador puede jugar dentro del torneo * Comprobar que, si el jugador pierde, ya no forma parte del torneo y no puede seguir jugando | | |
| **Observaciones:** | | |

|  |  |  |
| --- | --- | --- |
| **Identificador:** HU14 | Modificar datos personales | |
| **Descripción**: Como usuario quiero poder modificar mis datos personales | | |
| **Estimación** 2 | **Prioridad 3** | **Entrega** |
| **Pruebas de aceptación:**   * Introducir los campos vacíos y comprobar que se muestra un mensaje de error * Introducir los valores nuevos y comprobar que se han modificado en el sistema y en la interfaz del usuario | | |
| **Observaciones:** | | |

|  |  |  |
| --- | --- | --- |
| **Identificador:** HU15 | Ver ranking | |
| **Descripción**: Como jugador quiero poder ver el ranking | | |
| **Estimación** 5 | **Prioridad** 3 | **Entrega** |
| **Pruebas de aceptación:**   * Comprobar que al usuario se le muestra de forma ordenada por el criterio seleccionado los mejores jugadores * Comprobar que los jugadores mostrados están registrados en el sistema | | |
| **Observaciones:** | | |

|  |  |
| --- | --- |
| **Identificador:** HU16 | Cambiar filtro ranking |
| **Descripción**: Como jugador quiero poder seleccionar el filtro del ranking | |
| **Estimación** 2  **Prioridad** 3  **Entrega** | |
| **Pruebas de aceptación:**   * Comprobar que al introducir el nuevo filtro el ranking cambia * Comprobar que el usuario no puede introducir un filtro no válido | |

|  |  |
| --- | --- |
| **Identificador:** HU17 | Ver perfil jugador |
| **Descripción**: Como jugador quiero poder ver mi perfil de jugador | |
| **Estimación** 2  **Prioridad** 3  **Entrega** | |
| **Pruebas de aceptación:**   * Comprobar que al usuario se le muestran los valores almacenados en el sistema | |
| **Observaciones:** | |

|  |  |  |
| --- | --- | --- |
| **Identificador:** HU18 | Seleccionar idioma | |
| **Descripción**: Como usuario quiero poder cambiar el idioma de la plataforma | | |
| **Estimación** 2 | **Prioridad** 4 | **Entrega** |
| **Pruebas de aceptación:**   * Comprobar que al usuario se le muestran toda la plataforma en el idioma seleccionado * Comprobar que el selector de idioma ha cambiado y aparece el idioma seleccionado | | |
| **Observaciones:** | | |

|  |  |  |
| --- | --- | --- |
| **Identificador:** HU19 | Lista amigos | |
| **Descripción**: Como jugador quiero poder ver mi lista de amigos | | |
| **Estimación** 3 | **Prioridad** 2 | **Entrega** |
| **Pruebas de aceptación:**   * Comprobar que, si el usuario borra un amigo, éste ya no aparece en su lista de amigos * Comprobar que, si se conecta un amigo, este aparece con algún símbolo de que está conectado en la lista de amigos * Comprobar que los amigos de la lista son los que están almacenados en el sistema | | |
| **Observaciones:** | | |

|  |  |  |
| --- | --- | --- |
| **Identificador:** HU20 | Añadir amigo | |
| **Descripción**: Como jugador quiero poder buscar a otro jugador por su nombre y mandarle una solicitud de amistad | | |
| **Estimación** 3 | **Prioridad** 2 | **Entrega** |
| **Pruebas de aceptación:**   * Comprobar que al amigo ha recibido la solicitud de amistad * Comprobar que el nuevo amigo aparece en su lista de amigos * Comprobar que la búsqueda introducida y los usuarios registrados están relacionados * Comprobar que si el amigo ya está añadido a la lista de amigos que no se pueda mandar una solicitud de amistad | | | |
| **Observaciones:** | | | |

|  |  |  |
| --- | --- | --- |
| **Identificador:** HU21 | Aceptar solicitud amistad | |
| **Descripción**: Como jugador quiero poder aceptar una solicitud de amistad | | |
| **Estimación** 2 | **Prioridad** 2 | **Entrega** |
| **Pruebas de aceptación:**  • Comprobar que el usuario tiene al nuevo amigo añadido a la lista de amigos | | |
| **Observaciones:** | | |

|  |  |  |
| --- | --- | --- |
| **Identificador:** HU22 | Eliminar amigo | |
| **Descripción**: Como jugador quiero poder eliminar un jugador de mi lista de amigos | | |
| **Estimación** 1 | **Prioridad** 4 | **Entrega** |
| **Pruebas de aceptación:**   * Comprobar que el usuario eliminado ya no forma parte de la lista de amigos * Comprobar que a ese usuario se le puede enviar una solicitud de amistad | | |
| **Observaciones:** | | |

|  |  |
| --- | --- |
| **Identificador:** HU23 | Ver perfil amigo |
| **Descripción**: Como jugador quiero poder ver el perfil de jugador de un amigo | |
| **Estimación** 2  **Prioridad** 4  **Entrega** | |
| **Pruebas de aceptación:**   * Comprobar que al usuario se le muestran los atributos almacenados del jugador seleccionado | |
| **Observaciones:** | |

|  |  |  |
| --- | --- | --- |
| **Identificador:** HU24 | Cambiar modo visualización | |
| **Descripción**: Como usuario quiero poder cambiar el modo de visualización de la plataforma (modo predeterminado / modo oscuro) | | |
| **Estimación** 3 | **Prioridad** 5 | **Entrega** |
| **Pruebas de aceptación:**   * Comprobar que al usuario se le muestra toda la aplicación en el modo seleccionado * Comprobar que se ha cambiado el icono del modo de visualización al seleccionado. | | |
| **Observaciones:** | | |

|  |  |  |
| --- | --- | --- |
| **Identificador:** HU25 | Ver buzón | |
| **Descripción**: Como jugador quiero poder ver mis notificaciones a través del buzón | | |
| **Estimación** 2 | **Prioridad 2** | **Entrega** |
| **Pruebas de aceptación:**   * Comprobar que al usuario se le muestra todas las notificaciones recientes * Comprobar que el buzón no crece demasiado (no se eliminan antiguas notificaciones) | | |
| **Observaciones:** En el buzón se recibirán notificaciones de: solicitudes de amistad, solicitud de ingreso en un torneo, solicitudes de partidas 1vs1. | | |

Velocidad y planificación de Sprints: Wordle+.

Alumno: David Correa Rodríguez

Tutor: Juan Manuel Fernández Luna

***Velocidad***

Un aspecto que destacar de la metodología Scrum2 es que el equipo de desarrollo tiene asociado una **velocidad de trabajo**. Esta magnitud representa la cantidad de trabajo realizado en un periodo de tiempo. Esto quiere decir que, si el equipo tiene una velocidad de X puntos, el equipo realizará X puntos en cada sprint.

Este valor es importante a la hora de realizar la planificación de Sprints, ya que ésta influye directamente en qué historias de usuario van a ser completadas en qué Sprint en concreto.

Normalmente, para estimar este valor, se tiene en cuenta los Puntos de Historia (PH) completados en el primer Sprint. Como estos datos no están disponibles actualmente, se realizará una planificación inicial de Sprints aproximada, con un valor de velocidad máxima de 8PH. Dicha planificación estará sujeta a cambios, dependiendo de la velocidad calculada y de la evolución de los Sprints

***Planificación de Sprints.***

La planificación de los Sprints se ha realizado según los siguientes criterios

* Por orden de **prioridad**: las historias de usuario más prioritarias se realizarán en Sprints más tempranos
* Por agrupamiento de similitud: las historias de usuario similares se realizarán en los mismos Sprints.

De esta forma, la planificación queda de la siguiente manera:

|  |  |  |  |
| --- | --- | --- | --- |
| **Nº** | **Objetivo** | **Fecha de entrega** | |
| **1** | **Infraestructura del proyecto. Registro de jugadores** | **13 de febrero del 2023** | |
| **Ident.** | **Título** | **Estimación** | **Prioridad** |
| **HT** | Establecer la infraestructura del proyecto. | 4 | 1 |
| **HU.1** | Como jugador quiero poder registrarme | 2 | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nº** | **Objetivo** | **Fecha de entrega** | |
| **2** | **Registro de GE. Inicio y cierre de sesión** | **27 de febrero del 2023** | |
| **Ident.** | **Título** | **Estimación** | **Prioridad** |
| **HU.3** | Como usuario (jugador y gestor) quiero poder iniciar sesión | 2 | 1 |
| **HU.4** | Como usuario quiero poder cerrar sesión | 1 | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nº** | **Objetivo** | **Fecha de entrega** | |
| **3** | **Wordle clásico** | **13 de marzo del 2023** | |
| **Ident.** | **Título** | **Estimación** | **Prioridad** |
| **HU.5** | Como jugador quiero poder echar un Wordle clásico | 8 | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nº** | **Objetivo** | **Fecha de entrega** | |
| **4** | **Ampliación Wordle clásico. Modificación datos usuarios** | **27 de marzo del 2023** | |
| **Ident.** | **Título** | **Estimación** | **Prioridad** |
| **HU.6** | Como jugador quiero poder  seleccionar la longitud de la palabra de la partida clásica | 2 | 2 |
| **HU.14** | Como usuario quiero poder modificar mis datos personales | 2 | 3 |
| **HU.25** | Como jugador quiero ver las notificaciones a través del buzón | 2 | 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nº** | **Objetivo** | **Fecha de entrega** | |
| **5** | **Creación torneo** | **10 de abril del 2023** | |
| **Ident.** | **Título** | **Estimación** | **Prioridad** |
| **HU.7** | Como gestor de eventos quiero crear una sala abierta de un torneo | 5 | 2 |
| **HU.17** | Como jugador quiero poder ver mi perfil de jugador | 2 | 3 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nº** | **Objetivo** | **Fecha de entrega** | |
| **6** | **Configuración torneo** | **24 de abril del 2023** | |
| **Ident.** | **Título** | **Estimación** | **Prioridad** |
| **HU.8** | Como GE quiero poder seleccionar el tamaño del torneo (número de participantes) | 2 | 2 |
| **HU.9** | Como GE quiero poder seleccionar la longitud de la palabra del torneo | 2 | 2 |
| **HU.10** | Como GE quiero poder crear un torneo y preseleccionar los participantes | 3 | 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nº** | **Objetivo** | **Fecha de entrega** | |
| **7** | **Multijugador 1vs1** | **8 de mayo del 2023** | |
| **Ident.** | **Título** | **Estimación** | **Prioridad** |
| **HU.11** | Como jugador quiero poder jugar una partida 1vs1 con un amigo | 5 | 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nº** | **Objetivo** | **Fecha de entrega** | |
| **8** | **Inscripción al torneo** | **22 de mayo del 2023** | |
| **Ident.** | **Título** | **Estimación** | **Prioridad** |
| **HU.12** | Como jugador quiero poder seleccionar la longitud de la palabra del torneo (tipo de torneo) | 3 | 2 |
| **HU.13** | Como jugador quiero poder unirme a una sala abierta de un torneo | 2 | 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nº** | **Objetivo** | **Fecha de entrega** | |
| **9** | **Visualización del ranking** | **05 de junio del 2023** | |
| **Ident.** | **Título** | **Estimación** | **Prioridad** |
| **HU.15** | Como jugador quiero poder ver el ranking | 5 | 3 |
| **HU.16** | Como jugador quiero poder seleccionar el filtro del ranking | 2 | 3 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nº** | **Objetivo** | **Fecha de entrega** | |
| **10** | **Perfil de jugador. Solicitudes de amistad. Lista de amigos.** | **19 de junio del 2023** | |
| **Ident.** | **Título** | **Estimación** | **Prioridad** |
| **HU.19** | Como jugador quiero poder ver mi lista de amigos | 3 | 2 |
| **HU.20** | Como jugador quiero poder buscar a otro jugador por su nombre y mandarle  una solicitud de amistad | 3 | 2 |
| **HU.21** | Como jugador quiero poder aceptar una solicitud de amistad | 2 | 2 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nº** | **Objetivo** | **Fecha de entrega** | |
| **11** | **Eliminación de amigos. Perfiles de amigos. Cambio de idioma** | **03 de julio del 2023** | |
| **Ident.** | **Título** | **Estimación** | **Prioridad** |
| **HU.22** | Como jugador quiero poder eliminar un jugador de mi lista de amigos | 1 | 4 |
| **HU.23** | Como jugador quiero poder ver el perfil de jugador de un amigo | 2 | 4 |
| **HU.18** | Como usuario quiero poder cambiar el idioma de la plataforma | 2 | 4 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nº** | **Objetivo** | **Fecha de entrega** | |
| **12** | **Modo oscuro** | **17 de julio del 2023** | |
| **Ident.** | **Título** | **Estimación** | **Prioridad** |
| **HU.24** | Como usuario quiero poder cambiar el modo de visualización de la plataforma (modo predeterminado / modo oscuro) | 3 | 5 |

Esta planificación inicial muestra que el Sprint número 12 se quedaría fuera de plazo de la entrega del producto, siempre y cuando la velocidad del equipo no varíe. Si la velocidad de equipo se ve incrementada después de varios Sprints completados, éste último se incorporará al producto.

División en tareas: Wordle+.

Alumno: David Correa Rodríguez

Tutor: Juan Manuel Fernández Luna

Una vez definidas las Historias de Usuario, éstas se pueden dividir en tareas. Una tarea no es parte del resultado del proyecto, sino que es un medio para producir un resultado.

En este proyecto se seguirá un patrón *Modelo Vista Controlador*, y por ello, se pueden distinguir dos tipos de tareas:

* Tareas de **frontend:** son las tareas relacionadas con el diseño, implementación y desarrollo de las vistas de usuario e interfaces, es decir, con lo que el usuario podrá interactuar directamente.
* Tareas de **backend**: son las tareas relacionadas con la base de datos y el controlador. Son funcionalidades que el usuario no va a interactuar directamente con ellas, pero son necesarias para que el sistema, en su cómputo total, funcione correctamente.

|  |  |
| --- | --- |
| **Identificador:** HU1 | Registro jugador |
| |  |  | | --- | --- | | **Identificador** | **Título de la tarea** | | Tarea 1-1 | Definir el modelo de jugador en la BBDD | | Tarea 1-2 | Implementación la inserción automática del jugador en la BBDD | | Tarea 1-3 | Definir el modelo de usuario en la BBDD | | Tarea 1-4 | Implementación la inserción automática del usuario en la BBDD | | Tarea 1-5 | Definir el controlador asociado al registro del jugador | | Tarea 1-6 | Diseño e implementación de la IU asociada a la creación del jugador | | |
| **Observaciones:** | |

|  |  |
| --- | --- |
| **Identificador:** HU2 | Registro del GE |
| |  |  | | --- | --- | | **Identificador** | **Título de la tarea** | | Tarea 2-1 | Definir el modelo del GE en la BBDD | | Tarea 2-2 | Implementación la inserción automática del GE en la  BBDD | | Tarea 2-3 | Definir el controlador asociado al registro del GE | | Tarea 2-4 | Diseño e implementación de la IU asociada a la creación del GE | | |
| **Observaciones:**  - El modelo de usuarios ya está creado. | |

|  |  |
| --- | --- |
| **Identificador:** HU3 | Iniciar sesión |
| |  |  | | --- | --- | | **Identificador** | **Título de la tarea** | | Tarea 3-1 | Definir el controlador asociado al inicio de sesión | | Tarea 3-2 | Diseño e implementación de la IU asociada al inicio de sesión | | |
| **Observaciones:**  - El modelo de usuario está creado | |

|  |  |
| --- | --- |
| **Identificador:** HU4 | Cerrar sesión |
| |  |  | | --- | --- | | **Identificador** | **Título de la tarea** | | Tarea 4-1 | Definir el controlador asociado al cierre de sesión | | |
| **Observaciones:**  - El modelo de usuario está creado | |

|  |  |
| --- | --- |
| **Identificador:** HU5 | Wordle clásico |
| |  |  | | --- | --- | | **Identificador** | **Título de la tarea** | | Tarea 5-1 | Implementación de la lógica del Wordle clásico | | Tarea 5-2 | Diseño e implementación de la IU asociada al Wordle clásico | | |
| **Observaciones:** | |

|  |  |
| --- | --- |
| **Identificador:** HU6 | Longitud palabra Wordle clásico |
| |  |  | | --- | --- | | **Identificador** | **Título de la tarea** | | Tarea 6-1 | Definir el controlador asociado al cambio de longitud de la palabra | | Tarea 6-2 | Diseño e implementación de la IU asociada a la selección de la longitud de la palabra | | |
| **Observaciones:**  - El usuario, al querer jugar a un Wordle clásico, tendrá una elección previa que será el tamaño de la palabra del Wordle. | |

|  |  |
| --- | --- |
| **Identificador:** HU7 | Crear sala torneo |
| |  |  | | --- | --- | | **Identificador** | **Título de la tarea** | | Tarea 7-1 | Definir el modelo de la sala torneo en la BBDD | | Tarea 7-2 | Implementación la inserción automática de la sala del torneo en la BBDD | | Tarea 7-3 | Definir el controlador asociado a la creación de la sala del torneo | | Tarea 7-4 | Diseño e implementación de la IU asociada a la creación de la sala del torneo | | |
| **Observaciones:**  - Vista solo apta para GEs. | |

|  |  |
| --- | --- |
| **Identificador:** HU8 | Tamaño torneo |
| |  |  | | --- | --- | | **Identificador** | **Título de la tarea** | | Tarea 8-1 | Definir el controlador asociado al cambio de longitud de la palabra | | Tarea 8-2 | Diseño e implementación de la IU asociada a la selección de la longitud de la palabra | | |
| **Observaciones:**  - Vista apta solamente para GEs. | |

|  |  |
| --- | --- |
| **Identificador:** HU9 | Tamaño palabra torneo |
| |  |  | | --- | --- | | **Identificador** | **Título de la tarea** | | Tarea 9-1 | Definir el controlador asociado al cambio de tamaño de la palabra del torneo | | Tarea 9-2 | Diseño e implementación de la IU asociada a la selección de la longitud de la palabra del torneo | | |
| **Observaciones:**  - Vista apta solo para GEs. | |

|  |  |
| --- | --- |
| **Identificador:** HU10 | Crear torneo fijo |
| |  |  | | --- | --- | | **Identificador** | **Título de la tarea** | | Tarea 7-1 | Definir el controlador asociado a la creación del torneo | | Tarea 7-2 | Enviar notificación correspondiente a los jugadores seleccionados. | | Tarea 7-3 | Diseño e implementación de la IU asociada a la creación del torneo fijo | | |
| **Observaciones:**   * El modelo del torneo ya está creado. * Vista apta solo para GEs. | |

|  |  |
| --- | --- |
| **Identificador:** HU11 | Crear partida 1vs1 |
| |  |  | | --- | --- | | **Identificador** | **Título de la tarea** | | Tarea 11-1 | Definir el controlador asociado a la creación de la partida 1vs1 | | Tarea 11-2 | Diseño e implementación de la IU asociada a la partida 1vs1 | | Tarea 11-3 | Enviar notificación correspondiente al jugador. | | |
| **Observaciones:**   * Se completará cuando se implemente la lista de amigos. * El modelo del jugador ya está creado. | |

|  |  |
| --- | --- |
| **Identificador:** HU12 | Seleccionar longitud palabra torneo jugador |
| |  |  | | --- | --- | | **Identificador** | **Título de la tarea** | | Tarea 12-1 | Diseño e implementación de la IU asociada a la selección de la longitud de palabra en el torneo. | | |
| **Observaciones:**  - El usuario, al querer ingresar a un torneo, tendrá una elección previa que será el tamaño de la palabra de dicho torneo (tipo). | |

|  |  |
| --- | --- |
| **Identificador:** HU13 | Unirse a sala |
| |  |  | | --- | --- | | **Identificador** | **Título de la tarea** | | Tarea 13-1 | Definir el controlador asociado a la añadir al usuario a la sala seleccionada | | Tarea 13-2 | Diseño e implementación de la IU asociada a la unirse a la sala. | | |
| **Observaciones:**   * El modelo de las salas ya está creado. * El modelo de los jugadores ya está creado. | |

|  |  |
| --- | --- |
| **Identificador:** HU14 | Modificar datos personales |
| |  |  | | --- | --- | | **Identificador** | **Título de la tarea** | | Tarea 14-1 | Definir el controlador asociado a la modificación de los datos del usuario. | | Tarea 14-2 | Diseño e implementación de la IU asociada a la ficha del usuario. | | |
| **Observaciones:**  - El modelo de los usuarios ya está creado. | |

|  |  |
| --- | --- |
| **Identificador:** HU15 | Ver ranking |
| |  |  | | --- | --- | | **Identificador** | **Título de la tarea** | | Tarea 15-1 | Definir el controlador asociado a obtener los usuarios del ranking. | | Tarea 15-2 | Diseño e implementación de la IU asociada al ranking. | | |
| **Observaciones:**  - El modelo de los jugadores ya está creado. | |

|  |  |
| --- | --- |
| **Identificador:** HU16 | Cambiar filtro ranking |
| |  |  | | --- | --- | | **Identificador** | **Título de la tarea** | | Tarea 16-1 | Definir el controlador asociado a obtener los usuarios del ranking según el filtro escogido. | | |
| **Observaciones:** | |

|  |  |
| --- | --- |
| **Identificador:** HU17 | Ver perfil jugador |
| |  |  | | --- | --- | | **Identificador** | **Título de la tarea** | | Tarea 17-1 | Definir el controlador asociado a obtener la información del jugador. | | Tarea 17-2 | Diseño e implementación de la IU asociada al perfil del jugador. | | |
| **Observaciones:**  - El modelo de los jugadores ya está creado. | |

|  |  |
| --- | --- |
| **Identificador:** HU18 | Seleccionar idioma |
| |  |  | | --- | --- | | **Identificador** | **Título de la tarea** | | Tarea 18-1 | Definir el controlador asociado a cambiar el idioma de la aplicación | | |
| **Observaciones:** | |

|  |  |
| --- | --- |
| **Identificador:** HU19 | Lista amigos |
| |  |  | | --- | --- | | **Identificador** | **Título de la tarea** | | Tarea 19-1 | Definir el controlador asociado a obtener la lista de amigos. | | Tarea 19-2 | Diseño e implementación de la IU asociada a la lista de amigos. | | |
| **Observaciones:**  - El modelo de los jugadores ya está creado. | |

|  |  |
| --- | --- |
| **Identificador:** HU20 | Añadir amigo |
| |  |  | | --- | --- | | **Identificador** | **Título de la tarea** | | Tarea 20-1 | Definir el controlador asociado a añadir un amigo. | | Tarea 20-2 | Diseño e implementación de la IU asociada a añadir a un amigo (botón en lista de amigos) | | Tarea 20-3 | Enviar notificación correspondiente al jugador. | | |
| **Observaciones:**  - El modelo de los jugadores ya está creado. | |

|  |  |
| --- | --- |
| **Identificador:** HU21 | Aceptar solitud amistad |
| |  |  | | --- | --- | | **Identificador** | **Título de la tarea** | | Tarea 21-1 | Definir el controlador asociado a aceptar a un amigo. | | Tarea 21-2 | Diseño e implementación de la IU asociada a aceptar a un amigo (botón) | | Tarea 21-3 | Enviar notificación correspondiente al jugador. | | |
| **Observaciones:**  - El modelo de los jugadores ya está creado. | |

|  |  |
| --- | --- |
| **Identificador:** HU22 | Eliminar amigo |
| |  |  | | --- | --- | | **Identificador** | **Título de la tarea** | | Tarea 22-1 | Definir el controlador asociado a eliminar a un amigo | | Tarea 22-2 | Diseño e implementación de la IU asociada a eliminar a un amigo (botón) | | |
| **Observaciones:**  - El modelo de los jugadores ya está creado. | |

|  |  |
| --- | --- |
| **Identificador:** HU23 | Ver perfil amigo |
| |  |  | | --- | --- | | **Identificador** | **Título de la tarea** | | Tarea 22-1 | Definir el controlador asociado a ver el perfil de un amigo | | Tarea 22-2 | Diseño e implementación de la IU asociada a ver el perfil de un amigo (botón) | | |
| **Observaciones:**   * El modelo de los jugadores ya está creado. * La IU asociada a ver el perfil de un jugador ya está creada | |

|  |  |
| --- | --- |
| **Identificador:** HU24 | Cambiar modo visualización |
| |  |  | | --- | --- | | **Identificador** | **Título de la tarea** | | Tarea 24-1 | Definir el controlador asociado al cambio de visualización | | Tarea 24-2 | Diseño e implementación de la IU asociada al cambio de visualización (botón) | | |
| **Observaciones:** | |

|  |  |
| --- | --- |
| **Identificador:** HU25 | Ver buzón |
| |  |  | | --- | --- | | **Identificador** | **Título de la tarea** | | Tarea 25-1 | Definir el modelo de las Notificaciones en la BBDD | | Tarea 25-2 | Implementación la inserción automática de las notificaciones en la BBDD | | Tarea 25-3 | Definir el controlador asociado al buzón | | Tarea 25-4 | Diseño e implementación de la IU asociada al buzón. | | |
| **Observaciones:** | |

Tecnologías y herramientas: Wordle+.

Alumno: David Correa Rodríguez

Tutor: Juan Manuel Fernández Luna

***Diseño e implementación del proyecto***

***Frontend***

Una vez planteadas las HUs, su división en tareas, y la planificación de los Sprints, es necesario realizar un análisis de las tecnologías y herramientas candidatas para realizar este proyecto.

Antes de profundizar más en la cuestión, me gustaría aclarar que uno de los grandes objetivos de este proyecto es el aprendizaje y la adquisición de nuevo conocimiento, al igual que todo mi transcurso por la carrera, por lo que le daré mayor peso a aquellas herramientas con las que no tengo experiencia y/o demasiado conocimiento. A priori, sería más sencillo utilizar herramientas ya conocidas, pero estoy más interesado en ampliar mi perfil académico e invertir cierto tiempo en aprender ámbitos nuevos.

Como ya se ha mencionado varias veces, Wordle+ consiste en realizar un proyecto *fullstack,* es decir, tiene componentes tanto *backend* como *frontend*. Además, uno de los principales objetivos del proyecto es conseguir una aplicación web multiplataforma, por lo que tecnologías dedicadas exclusivamente para desarrollo de aplicaciones para móviles (como Android Studio[[5]](#endnote-5), Flutter[[6]](#endnote-6), etc) quedarán descartadas. Con esto, el dominio queda reducido a tecnologías web, por lo que las alternativas principales son:

* **Pila LAMP[[7]](#endnote-7)**: consiste en una infraestructura que utiliza Linux[[8]](#endnote-8), Apache[[9]](#endnote-9), MySQL/MariaDB[[10]](#endnote-10) y PHP[[11]](#endnote-11), también utilizando JavaScript[[12]](#endnote-12) como lenguaje de *scripting*. Es una infraestructura ya ampliamente utilizada en otros proyectos de la carrera, por lo que quedan descartados. Además, puede resultar complicado diseñar completamente el proyecto de forma adaptativa o *responsive.*
* **Pila MERN/MEAN[[13]](#endnote-13):** similar a la pila LAMP**7**, pero o se utiliza MongoDB[[14]](#endnote-14) como gestor de base de datos; Express como gestor de peticiones HTTP; React[[15]](#endnote-15) o Angular[[16]](#endnote-16) como herramienta principal para el *frontend;* y Node.js[[17]](#endnote-17) como plataforma de *backend*. El principal lenguaje a utilizar es JavaScript12. Esta alternativa resulta interesante y se presenta como una renovación de la pila LAMP**7**. Sin embargo, personalmente ya he utilizado esta infraestructura (MERN**13**) en un proyecto de la asignatura “Dirección y Gestión de Proyectos”. En enlace al repositorio público es: <https://github.com/davidcr01/Class4All>.Además, presenta el mismo problema del diseño *responsive* que presentaba la pila LAMP**7**.

Con lo anterior expuesto, se podría cambiar el foco a frameworks que permitan el desarrollo de aplicaciones multiplataforma o híbridas. A pesar de que hay una extensa lista de alternativas, se comentarán las principales de ellas:

* **Flutter**6: si bien es cierto que ha sido mencionado anteriormente, Flutter6 es uno de los sistemas más utilizados para el desarrollo de este tipo de aplicaciones. Sin embargo, se suele utilizar en mayor medida para el desarrollo de aplicaciones móviles, además de utilizar el lenguaje de programación Dart[[18]](#endnote-18), por lo que se aleja del uso de tecnologías web, que es uno de los objetivos de este proyecto.
* **React Native[[19]](#endnote-19):** se presenta como otra alternativa interesante, en donde el lenguaje principal es JavaScript, con posibilidad de escribir módulos en Objective-C[[20]](#endnote-20), Swift[[21]](#endnote-21) o Java[[22]](#endnote-22). Sin embargo, como ya he comentado anteriormente, he realizado un proyecto utilizando React15, por lo que el aprendizaje que me aportaría esta alternativa se vería reducido.
* **Ionic Framework[[23]](#endnote-23):** al igual que las demás, permite crear aplicaciones híbridas, pero en este caso utilizando HTML, CSS y JavaScript12, con otra notación distinta. Este framework permite un desarrollo ágil para diseñar aplicaciones híbridas, pudiendo utilizar elementos web ya construidos. Cabe destacar que es uno de los framework más utilizado actualmente, además de tener una gran comunidad y documentación.

Finalmente he escogido de **Ionic Framework23**, por lo comentado anteriormente. Sigue la línea de las tecnologías web, que es uno de los objetivos de este proyecto, permitiendo desarrollar Wordle+ tanto en web como en móvil de forma **simultánea.** Además, esta alternativa permite utilizar tres de los grandes lenguajes de programación *frontend* actual a elegir, es decir, solamente se puede utilizar una de ellas: React15, Angular16 y Vue[[24]](#endnote-24). En mi caso, descarto la primera por haberla utilizado en otros proyectos, y respecto a las dos restantes, ambas son similares, pero **Angular**16 tiene una comunidad y tasa de mercado mucho más significativa.

Por tanto, y a modo de resumen, se utilizará el framework Ionic**23** utilizando el lenguaje de programación Angular16.

***Backend***

El groso de las tecnologías comentadas anteriormente son herramientas de desarrollo de *frontend*. Respecto al *backend*, podemos categorizar las alternativas siguiendo diferentes criterios:

* Por **cómo** se gestiona la información: las principales alternativas en este campo son modelos **relacionales** y modelos **no relacionales**. Personalmente he utilizado ambos exhaustivamente, con herramientas como MySQL10 y MongoDB14.   
  Sin embargo, tengo cierto interés en utilizar **PostgreSQL[[25]](#endnote-25)**, uno de los sistemas de gestión de bases de datos relacional más utilizado actualmente. En este caso, tengo preferencia en utilizar modelos relacionales, ya que gran parte de la información que va a alojar la base de datos tiene bastante conexión mediante claves externas, y tendría poco sentido utilizar modelos no relacionales.
* Por **dónde** se ubica la información: en este caso, las principales alternativas son utilizar servicios en la **nube** o alojar la base de datos de forma **local**. La nube es interesante ya que los datos están almacenados en internet, y hay poca probabilidad de pérdida. Sin embargo, se requiere de un proveedor externo, de un ancho de banda considerable para obtener la información, y evidentemente de conexión a internet.  
  Por otro lado, la alternativa local no presenta las desventajas descritas anteriormente, pero la información solamente estaría alojada en el sistema local. Finalmente, he preferido escoger la alternativa **local,** aunque para paliar sus desventajas me gustaría realizar un despliegue del backend en un contenedor Docker28, punto que se tratará más adelante.

Este proyecto presenta una arquitectura del tipo Modelo-Vista-Controlador. Ya se han mencionado los aspectos del “Modelo” y de la “Vista”. Respecto al controlador, se propone desarrollar una API REST, que permite la estandarización de intercambio de datos entre los servicios web, abstrayéndose del lenguaje de programación y de los sistemas operativos.

Utilizar una API REST tiene las siguientes ventajas:

* Escalabilidad: se optimizan las interacciones entre el cliente y el servidor.
* Flexibilidad: existe un desacoplo entre los componentes de manera que éstos pueden evolucionar de forma independiente
* Independencia: no importa la tecnología utilizada, no afecta al diseño de la API.

En cuanto a tecnologías, existen dos alternativas ampliamente conocidas:

* **Node.js**17: junto con la herramienta Express, es uno de los métodos más utilizados para crear APIs. Sin embargo, esta alternativa ya ha sido utilizada en el proyecto anteriormente mencionado.
* **Python[[26]](#endnote-26)**: uno de los lenguajes multiparadigma más utilizados en el panorama actual, permite construir APIs de forma sencilla mediante el framework **Django REST framework[[27]](#endnote-27)**. Esta opción me parece interesante para incorporar al proyecto una potente tecnología como es Python**26**, y hacer uso de otro de los frameworks más utilizados actualmente, como es el citado anteriormente.

***Abstracción del sistema***

Actualmente, la tecnología está evolucionando continuamente y a una gran velocidad. Ya se han mencionado anteriormente aspectos como la integración y distribución continuas (CICD4) que aportan nuevo software rápidamente, con actualizaciones muy frecuentes.

Ante este panorama, no es recomendable hacer un despliegue en un sistema en concreto, utilizando un sistema operativo y un hardware específico. Esta metodología haría el proyecto muy rígido, permitiendo no poder migrarlo en caso de que fuera necesario. Ante esto, surgen varios proyectos como **Docker****[[28]](#endnote-28)** que permiten realizar un despliegue de aplicaciones dentro de contenedores software. Esto, a su vez, abstrae y automatiza la virtualización de aplicaciones en distintos sistemas operativos.

Por ello, lo ideal sería hacer el despliegue de todo el proyecto utilizando la herramienta *docker-compose* que permite orquestar varios contenedores y realizar una comunicación interna entre ellos. Si esto no es posible, se pretende como mínimo desplegar el *backend* en contenedores, para así abstraer toda la información e instalación de esta parte del proyecto del sistema sobre el que se construya.

***Control de versiones y seguimiento del Sprint***

GitHub es sin duda alguna el sitio web en donde más desarrolladores alojan sus proyectos software. Gracias a su integración con la herramienta *Git* para el control de cambios, y sus características como las *Issues, Pull Request, Milestones*, la convierte en una herramienta idónea para este tipo de proyectos. Por tanto, el proyecto se alojará en un repositorio de GitHub, haciendo uso de Git para el control de cambios.

Respecto al seguimiento de los Sprints, existen varias conocidas alternativas como Trello o Jira, en donde plasmar todo el Product Backlog, planificación de los Sprints y su seguimiento. Sin embargo, esta tarea se puede también realizar el GitHub de formas distintas.

El sistema propuesto es crear varios hitos o *milestones* en GitHub, que representen las funcionalidades del proyecto, como por ejemplo “registro de usuarios”, “login de usuarios”, “Wordle clásico”, y a dichos hitos asociarles las respectivas Issues y Pull Request. De esta manera, se permite un seguimiento mucho más óptimo y una documentación más extensa del progreso del proyecto, teniendo en un mismo lugar los cambios realizados en el proyecto y su seguimiento asociado. Esta información ha sido extraída de: <https://docs.github.com/es/issues/using-labels-and-milestones-to-track-work/about-milestones>

Diseño de la Base de Datos: Wordle+.

Alumno: David Correa Rodríguez

Tutor: Juan Manuel Fernández Luna

***Análisis del contexto***

Para realizar un correcto diseño de la base de datos, previamente a caracterizar éste a la tecnología a utilizar, es necesario hacer un análisis de toda la información necesaria a almacenar y sus respectivas relaciones.

Para ello, y recapitulando todas las características principales del proyecto, existen varias entidades básicas de las que tenemos que almacenar información:

* Usuarios
* Partidas
* Torneos

De los usuarios derivan: los jugadores, los gestores de eventos, la lista de amigos, las peticiones de amistad y el buzón.

De los torneos derivan: las participaciones de los jugadores en los torneos y las rondas de cada torneo.

Con esta información, podemos identificar las identidades y las relaciones que se dan en la base de datos, añadiendo también los campos asociados:

* **Usuarios (entidad)**: nombre de usuario, correo, nombre, apellidos, fecha de registro, si es manager o no, y el avatar):
  + **Jugadores (entidad)**: nombre de usuario, wordles completados, partidas 1vs1 ganadas, torneos ganados, experiencia y categoría)
  + **Gestores de eventos (entidad)**: nombre de usuario, correo, nombre, apellidos, fecha de registro
* **Lista de amigos (relación)**: nombre de usuario de un jugador, nombre de usuario de otro jugador.
* **Solicitudes de amigos (entidad):** nombre de usuario del remitente, nombre de usuario del destinatario, la fecha de creación y si es válida o no.
* **Buzón (entidad):** nombre de usuario asociado al buzón, texto asociado y link asociado. Como no existe una clave primaria, esta entidad es una entidad débil de *Jugador*
* **Partidas (entidad):** identificador de la partida, nombre de usuario del jugador 1, nombre de usuario del jugador 2, si el jugador 1 ha jugado, si el jugador 2 ha jugado, y quién es el ganador.

Nota: las partidas serán asíncronas, es decir, no es necesario que ambos jugadores estén al mismo tiempo jugando la partida, por lo que es necesario los campos de si cada jugador ha jugado la partida.

* **Torneos (entidad):** identificador del torneo, nombre, descripción y número máximo de jugadores.
* **Participaciones (relación Jugadores-Torneos):** identificador del torneo asociado, nombre de usuario del jugador asociado.
* **Rondas (relación):** identificador de la ronda, identificador del torneo asociado, número de la ronda, partida asociada a dicha ronda

En el análisis podemos notar lo siguiente:

* Las entidades **Jugadores** y **Gestores** **de** **eventos** son entidades heredadas de **Usuarios**, por lo que tendrán una relación de herencia *es-un.*
* La entidad **Gestores de eventos** resulta innecesaria al tener un subconjunto (y solamente dicho subconjunto) de los campos de la entidad **Usuarios**.
* Existen muchas más relaciones sin mencionar, pero que son innecesarias y no corresponden a la base de datos, pero sí se tendrán en cuenta en el diagrama E/R[[29]](#endnote-29).

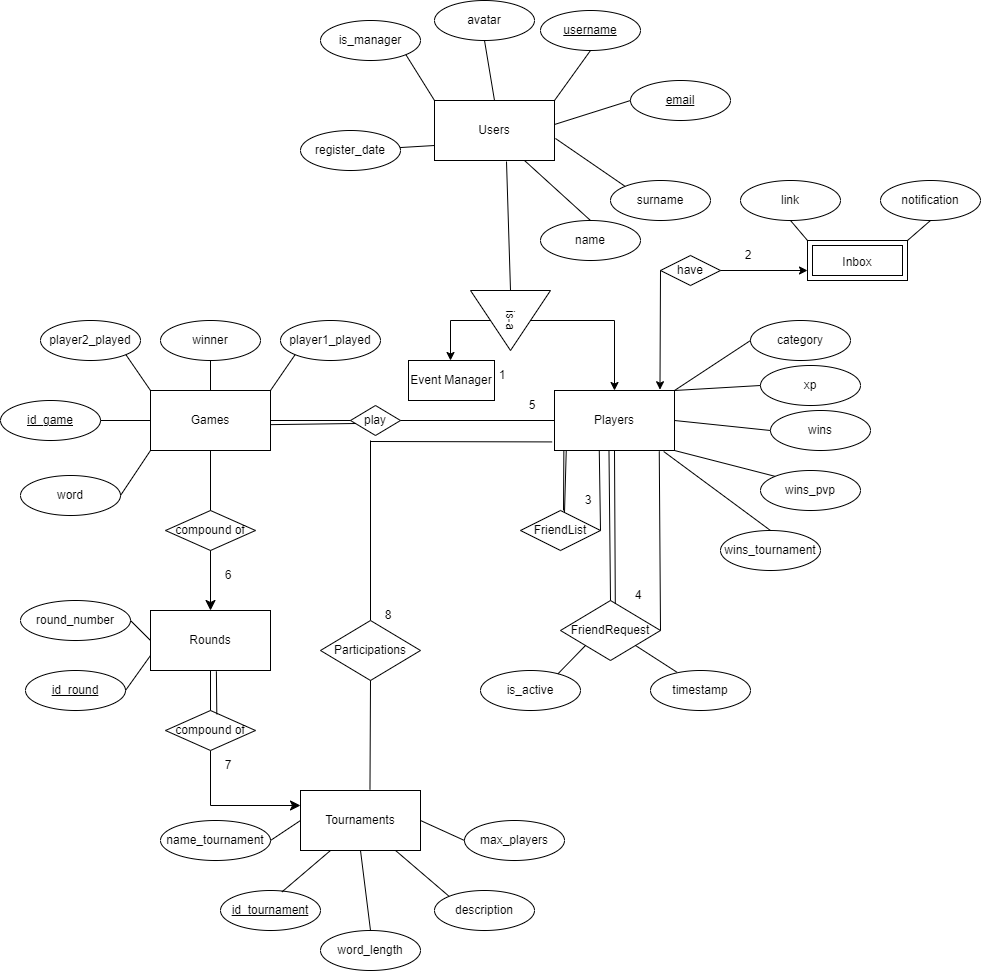
Estás relaciones pueden ser: *Jugador-Buzón* (tiene), *Jugador-Lista de amigos* (tiene), *Jugador-Solicitudes* de amistad (tiene), *Torneos-Rondas* (se componen de), *Jugador-Partidas* (juegan*), Rondas-Partidas* (se componen de).

* Existen varias entidades que dependen existencialmente de otras, es decir, la existencia de una entidad es obligatoria para la existencia de otra entidad.

Estas obligatoriedades se dan entre (la entidad de la izquierda es la que depende existencialmente de la otra): *Lista de amigos-Jugador, Solicitudes de amistad-Jugador, Buzón-Jugador, Partidas-Jugador, Ronda-Torneo*.

***Diagrama Entidad-Relación***

Teniendo en cuenta todos los aspectos destacados en el previo análisis, un posible diagrama Entidad-Relación sería el siguiente:

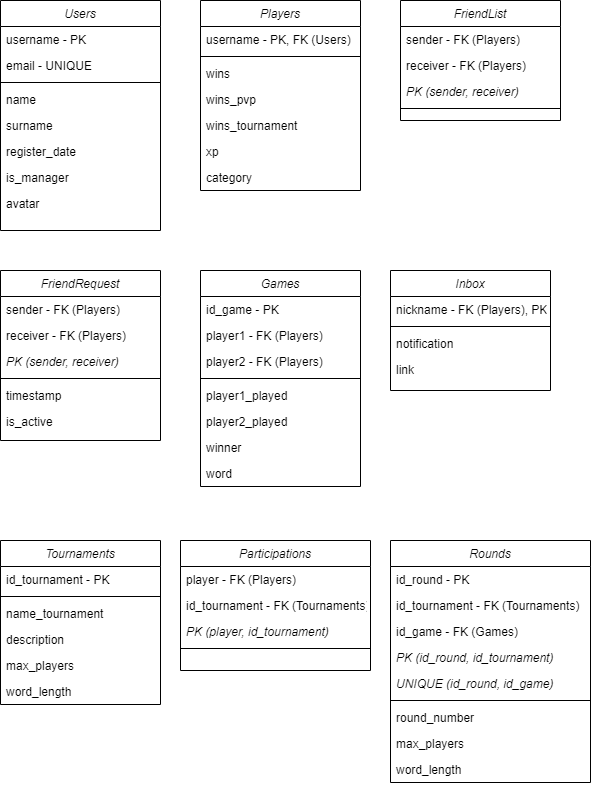


Algunas aclaraciones a tener en cuenta, numeradas en el diagrama, son:

1. Como se ha comentado anteriormente, la entidad *Event Manager* es una especialización de la entidad *Users* sin ningún atributo extra, por lo que añadiendo el campo *is\_manager* a la entidad *Users* no es necesario crear una tabla nueva para *Event Manager.*
2. Un jugador solamente tiene un buzón, y dicho buzón solamente puede pertenecer a un usuario: cardinalidad 1:1. Además, la entidad buzón es débil al no tener una clave primaria propia. Se ha obviado la obligatoriedad entre estas entidades al ser *Inbox* una entidad débil, por lo que dicha condición ya se cumple. La entidad *Inbox* heredará como clave externa y primaria la clave de la entidad *Player.*
3. Un jugador puede ser amigo de muchos jugadores, y también viceversa, por lo que presenta una cardinalidad N:M. Además, la tabla resultante de esta relación presenta una obligatoriedad con la entidad *Players* ya que su existencia depende de ésta. Esta relación heredará las claves primarias de la entidad *Players.*
4. Relación similar a la anterior, pero añadiendo atributos extra. Un jugador puede tener varias peticiones de amistad, y una misma petición de amistad puede pertenecer a varios jugadores (N:M), pero en este caso solamente puede pertenecer a uno de los dos jugadores, pero no de forma simultánea. Esta condición deberá comprobarse antes de introducir valores a la tabla resultante. La tabla resultante de esta relación exige una obligatoriedad con la entidad *Players* con la misma justificación que el punto anterior.
5. Un jugador puede jugar muchas partidas, y esa partida puede ser jugada varias veces por los mismos jugadores (repetir una partida), por lo que la relación presenta una cardinalidad N:M, con obligatoriedad de las partidas hacia los jugadores, ya que sin jugadores las partidas no pueden ser jugadas.
6. Una ronda está compuesta de varias partidas, pero una partida de un torneo en concreto solamente puede pertenecer a una ronda, ya que en un torneo no se repiten enfrentamientos entre los mismos jugadores en diferentes momentos (en torneos eliminatorios), por lo que la relación presenta una cardinalidad (1:N). No se ha especificado obligatoriedad en este caso ya que se permiten que haya rondas sin partidas, ya que no es posible saber las partidas de las rondas siguientes sin los resultados de la ronda actual en ese momento.
7. Un torneo se compone de varias rondas, pero dicha ronda solamente puede pertenecer a un torneo en concreto, por lo que la relación presenta una cardinalidad (1:N). Además, existe obligatoriedad entre las rondas y los torneos, ya que una ronda no tiene sentido sin antes existir un torneo que la contenga.
8. Un jugador puede participar en muchos torneos, y en un mismo torneo pueden participar muchos jugadores, por lo que la relación presenta una cardinalidad N:M.

***Paso a tablas***

En el modelo entidad relación se observan muchas entidades y relaciones, pero no es necesario representar todas las relaciones y entidades expuestas en el paso a tablas, por lo que las más importantes y fundamentales son:



Bibliografía y referencias: Wordle+.

1. Wikipedia. *es.wikipedia.org.* 22 de febrero de 2023. https://es.wikipedia.org/wiki/Desarrollo\_%C3%A1gil\_de\_software. [↑](#endnote-ref-1)
2. SA, ERIKA. *atlassian.com.* s.f. https://www.atlassian.com/es/agile/scrum/scrum-metrics (último acceso: 2 de febrero de 2023). [↑](#endnote-ref-2)
3. Kabanize. *¿Qué es Kanban?* s.f. https://kanbanize.com/es/recursos-de-kanban/primeros-pasos/que-es-kanban#:~:text=Kanban%20es%20un%20m%C3%A9todo%20Lean,la%20eficiencia%20y%20mejorar%20continuamente. [↑](#endnote-ref-3)
4. Unity. *¿Qué es CICD?* s.f. https://unity.com/es/solutions/what-is-ci-cd#:~:text=La%20CI%2FCD%2C%20o%20integraci%C3%B3n,la%20entrega%20continua%20de%20c%C3%B3digo. [↑](#endnote-ref-4)
5. Google. *developer.android.com.* s.f. https://developer.android.com/studio (último acceso: 25 de febrero de 2023). [↑](#endnote-ref-5)
6. Flutter. *https://flutter.dev/.* s.f. https://flutter.dev/ (último acceso: 25 de febrero de 2023). [↑](#endnote-ref-6)
7. IBM. *www.ibm.com.* 9 de mayo de 2019. https://www.ibm.com/es-es/cloud/learn/lamp-stack-explained. [↑](#endnote-ref-7)
8. XenForo. *linux.org.* s.f. https://www.linux.org/.

   9 Apache. *httpd.apache.org/.* s.f. https://httpd.apache.org/ (último acceso: 25 de febrero de 2023). [↑](#endnote-ref-8)
9. [↑](#endnote-ref-9)
10. Oracle. *mysql.com.* s.f. https://www.mysql.com/ (último acceso: 25 de febrero de 2023). [↑](#endnote-ref-10)
11. PHP. *php.net.* s.f. https://www.php.net/ (último acceso: 25 de febrero de 2023). [↑](#endnote-ref-11)
12. JavaScript. *javascript.com.* s.f. https://www.javascript.com/ (último acceso: 25 de febrero de 2023). [↑](#endnote-ref-12)
13. MongoDB. *What is the MERN stack?* s.f. https://www.mongodb.com/mern-stack#:~:text=MERN%20stands%20for%20MongoDB%2C%20Express,MongoDB%20%E2%80%94%20document%20database. [↑](#endnote-ref-13)
14. Inc., MongoDB. *mongodb.com.* s.f. https://www.mongodb.com/ (último acceso: 25 de febrero de 2023). [↑](#endnote-ref-14)
15. Platforms, Meta. *reactjs.org.* s.f. https://es.reactjs.org/. [↑](#endnote-ref-15)
16. Angular. *angular.io.* s.f. https://angular.io/ (último acceso: 25 de febrero de 2023). [↑](#endnote-ref-16)
17. Foundation, OpenJS. *nodejs.org.* s.f. https://nodejs.org/en/ (último acceso: 25 de febrero de 2023). [↑](#endnote-ref-17)
18. Dart. *dart.dev.* s.f. https://dart.dev/ (último acceso: 25 de febrero de 2023). [↑](#endnote-ref-18)
19. Native, React. *https://reactnative.dev/.* s.f. [↑](#endnote-ref-19)
20. Wikipedia. *Objective-C definition.* s.f. https://es.wikipedia.org/wiki/Objective-C. [↑](#endnote-ref-20)
21. Inc, Apple. *swift.* s.f. https://www.apple.com/es/swift/. [↑](#endnote-ref-21)
22. Oracle. *java.com.* s.f. https://www.java.com/es/ (último acceso: 25 de febrero de 2023). [↑](#endnote-ref-22)
23. Ionic. *https://ionicframework.com/.* s.f. [↑](#endnote-ref-23)
24. You, Evan. *vuejs.org.* s.f. https://vuejs.org/ (último acceso: 25 de febrero de 2023). [↑](#endnote-ref-24)
25. PostgreSQL. *postgresql.org.* s.f. https://www.postgresql.org/ (último acceso: 25 de febrero de 2023). [↑](#endnote-ref-25)
26. Python. *python.org.* s.f. https://www.python.org/ (último acceso: 25 de febrero de 2023). [↑](#endnote-ref-26)
27. Framework, Django Rest. *django-rest-framework.org.* s.f. https://www.django-rest-framework.org/ (último acceso: 25 de febrero de 2023). [↑](#endnote-ref-27)
28. Docker. *docker.com.* s.f. https://www.docker.com/ (último acceso: 25 de febrero de 2023). [↑](#endnote-ref-28)
29. Wikipedia. *Modelo entidad-relacion.* s.f. https://es.wikipedia.org/wiki/Modelo\_entidad-relaci%C3%B3n (último acceso: 25 de febrero de 2023). [↑](#endnote-ref-29)