

# Documento de seguimiento proyecto

## Conseguir los requisitos del nivel señalado.

### Justificación.

La planificación de esta entrega se recoge todas las historias de usuario que permiten realizar al user-player todas las acciones previas a una partida:

- Registro, inicio sesión y bases.
- Perfiles de usuario, invitación de amigo y gestión de invitaciones.
- Crear un lobby previo al matchmaking, invitación de cualquier usuario al lobby, invitación de amigos al lobby y gestión de invitaciones al lobby.
- Matchmaking con websocket que creara *Búsquedas de lobby* o *LobbySearch* donde los diferentes lobbies de diferentes usuarios puedan combinarse de manera condicionada.
- Creación de jugadores desde cada usuario del lobby, y creación de partida
- Creación de tableros de Acción, Guerra y Estados del a ciudad

### Registro, inicio de sesión y bases.

Antes de trabajar cada vista individual, se decidió realizar un diseño base que se compartiera entre todas estas y que permitiera tener componentes comunes que luego ayudaran con el responsive o a la reutilización. Esta parte incluye la vista welcome, que según el usuario hubiera iniciado sesión o no, mostraría botones para iniciar sesión/registrarse, o empezar partida o entrar en las futuras estadísticas del perfil. También se trabajó en la estructura de vista común, que ha sido utilizada por en todo el proyecto por las diferentes vistas de cada grupo de entidad-controlador-vista-repositorio.

La funcionalidad de inicio de sesión se basa en el funcionamiento que trae el ejemplo de Petclinic de Spring. Declarado en el archivo de configuración de Spring, se permite el uso del controlador de inicio que usa los token csrf. Esta parte se estudió por parte de un subgrupo para conseguir integrarlo en una vista no predeterminada por spring, pero tuvo que cesar esa parte del trabajo debido a la gran inversión de tiempo que estaba conllevando. El registro se realiza en el modelo de usuarios. Se crea un nuevo controlador, y se añaden nuevos métodos al servicio que utilizan el repositorio para guardar y recoger los usuarios. Además se realiza una vista customizada de formulario de registro.

### Perfiles de usuario, invitación de amigo y gestión de invitaciones

Los perfiles de usuario utilizan la base que venía con el ejemplo de spring, pero se modifican sus funcionalidades, la vista y se integran nuevos métodos para que un usuario que entra en un perfil que no es suyo, pueda enviar petición de amistad o aceptarla. También se tiene gestión de invitaciones a partida o de amistades desde el menú lateral.

Crear un lobby previo al matchmaking, invitación de cualquier usuario al lobby, invitación de amigos al lobby y gestión de invitaciones al lobby

Para empezar una partida, se requiere que un grupo de jugadores esté reunido. Se ofrece el modelo de Lobby para esta tarea. Cuando un usuario hace click en crear partida, realmente crea un nuevo lobby donde puede invitar a cualquier usuario o usuarios amigos desde el menú lateral. Un usuario puede aceptar invitación de lobby desde el mismo menú lateral, o desde la pantalla de welcome habiendo iniciado sesión.

Matchmaking con websocket que creará *Búsquedas de lobby* o *LobbySearch* donde los diferentes lobbies de diferentes usuarios puedan combinarse de manera condicionada

Una vez que desde el lobby se busca partida, se crea en memoria una búsqueda de lobby (*LobbySearch*) que empieza partida si está *satisfecha* o busca combinarse con otros lobbies si se encuentran dentro de su misma condición. Todos los usuarios dentro de lobby están escuchando el canal del websocket para escuchar y recoger las modificaciones de su propio lobby. Una partida se considera satisfecha (*satisfy*) si la cantidad de usuarios restantes menos la cantidad de usuarios que ya hay en la búsqueda es igual a 0. Dado este caso, se crea una lista de jugadores (*Players*) por cada usuario (*User*) del lobby, y se crea una partida que relaciona estos jugadores y los diferentes tableros.

Creación de jugadores desde cada usuario del lobby, y creación de partida

Tal y como se detalla en el anterior punto, es el matchmaking el encargado de crear una partida con los jugadores. No era objetivo de esta planificación integrar las funcionalidades dentro de la partida con los tableros, turnos y jugadores, pero se deja creada y lista para que puedan realizar las acciones correspondientes de las partidas y del resto de historia de usuario en la partida.

Creación de tableros de Acción, Guerra y Estados de la ciudad

Cuando se crea la partida, se crean los 3 tableros de la partida. Estos van ligados directamente a la partida e individualmente tienen una interfaz, algunos en canvas, otros directamente en HTML y Javascript, dentro de una estructura común realizada dentro de un tag de layout exclusivo para las partidas.

# Análisis retrospectivo del sprint. Reflexión.

## General

La mayor dificultad encontrada en este segundo sprint la hemos encontrado en trabajar en Spring, que nos ponía ciertas limitaciones con algunas funcionalidades, haciéndonos emplear algo de tiempo vanamente.

A pesar de seguir teniendo problemas en este ámbito, hemos mejorado nuestro rendimiento con respecto al primer sprint, con la organización por parejas jugando un factor clave en esto.

Finalmente consideramos que hemos conseguido un resultado satisfactorio, hemos estado a la altura de nuestra planificación, y nos encontramos motivados a seguir con la planificación del tercer sprint.

## Álvaro Vázquez - Gonzalo Martínez: Pantalla de inicio, Log In, registrarse y tablero de guerra.

En el inicio del sprint nos dedicamos ambos a diseñar en css y jsp la pantalla de inicio. Tras partir del diseño base, lo adaptamos a la pantalla de iniciar sesión y registrarse siendo ambos responsive para todo tipo de dispositivos. Esto nos dio bastantes problemas, los cuales aún persisten. Investigar cómo funciona el procesamiento de iniciar sesión en spring añadiendo una vista personalizada, nos llevó demasiado tiempo y no acabó de funcionar.

Tras acudir a la revisión el viernes, se nos indicó que no era importante, así que a partir de ese momento rectificamos, priorizamos tareas y nos dedicamos a la parte del tablero de guerra. Para ello, creamos controladores, las entidades, servicios y repositorios pertinentes así como la vista del propio siendo también responsive para todos dispositivos. Esta última parte fué la más costosa a implementar debido a dos problemas que surgieron durante la programación de los mismos, el primero fue el orden temporal en el que se ejecutaban las funciones definidas en los “.tag” lo que provocó que o bien el tablero o las fichas del mismo no se imprimiesen en su debido orden (se dibujaban en el orden en el que se cargasen las imágenes en el navegador). El segundo problema que surgió fue la mezcla de lenguajes (JSP y JavaScript) durante la programación de la integración de las fichas en el tablero, ya que los bucles se comportaban de manera independiente en cada lenguaje.

## Luis Rodríguez - Pedro Parrilla: Módulo social, panel de navegación lateral y aceptación de partidas.

Primeramente, nos dedicamos a la creación de la tabla “Amigos” y a desarrollar el servicio asociado (sin controlador debido a que los amigos son mostrados en el panel lateral y en los detalles de los usuarios). El principal problema era a la hora de diseñar la tabla ya que las relaciones de amistad son bidireccionales (X es amigo de Y tanto como Y es amigo de X) pero las peticiones no (la petición que envía X a Y, no es una petición de Y a X) de forma que teníamos que decidir entre unificar toda la “Amistad” en una tabla o tablas distintas de amistad y petición. Optamos por solo una tabla de “Amistad” en la que con un parámetro

booleano detectamos si es una petición de amistad y por otro lado según quién envía la petición de amistad, se detecta si es el origen o el destino de la petición. Una vez aceptada, el parámetro booleano cambia y los usuarios pertenecientes a esta relación siguen siendo “Origen” y “Destino”, aunque ya no importe el orden.

Las queries eran más complicadas en esta tabla debido a la bidireccionalidad pero una vez que se definieron bien, no hubo mayor problema.

Para la barra lateral, no hubo mayor problema, simplemente los módulos que se habían ido añadiendo y añadir un menú de navegación para hacer todo más cómodo ya que principalmente para movernos por la aplicación usábamos directamente la uri.

## Daniel Díaz - Álvaro Miranda: Lobby, invitaciones de partida, matchmaking y tablero de acciones.

En primer lugar, nos pusimos con el lobby. Empezamos enlazándolo con el botón de jugar del “welcome”, y a partir de ahí, fuimos construyéndolo. Configuramos las opciones por defecto y añadimos el panel para modificarlas, (el cual solo podía editarlo el administrador del lobby) y una lista de los jugadores del lobby.

Cuando terminamos la base, implementamos la opción de invitar a otros usuarios. Al trabajar en paralelo, teníamos que esperar a que se terminara el módulo social para probar las invitaciones a usuarios, por lo que probamos primero a invitar a un usuario cualquiera escribiendo su nombre, y una vez se terminó el módulo social, implementamos las invitaciones a amigos. Mientras Álvaro implementó dichas invitaciones, Daniel desarrolló el matchmaking para poder empezar partida una vez se llegara a los jugadores configurados. Finalmente, Daniel desarrolló el prototipo y funcionalidades básicas del tablero de acciones, siendo este la interfaz al comenzar partida.

Tipo	Gonzalo Martínez	Álvaro Vázquez	Luis Rodríguez	Álvaro Miranda	Daniel Díaz	Pedro Parrilla
Registro y login, pantalla inicio	23	23	6	-	-	-
Lobby	-	-	-	22	12	-
Amistad	-	-	18	15	-	19
Usuarios	-	-	16	-	-	13

Panel social	-	-	-	3	-	8
Matchmaking	-	-	-	-	19	-
Tableros	17	17	-	-	9	-
<b>Total:</b>	<b>40</b>	<b>40</b>	<b>40</b>	<b>40</b>	<b>40</b>	<b>40</b>

*La unidad de medida son las horas de trabajo.*