



Universidad
Zaragoza

Trabajo Fin de Grado

BacterField: Un videojuego de estrategia para
controlar el micromundo

BacterField: A strategy videogame to rule the
microworld

Autor

Ignacio Gracia Quílez

Director

Eduardo Mena Nieto

Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación

ESCUELA DE INGENIERÍA Y ARQUITECTURA

NOVIEMBRE 2021

*Dedicado a mi madre y mi hermana,
que siempre me han acompañado y han estado ahí.*

Agradecimientos

Agradecer a mi director en este trabajo, Eduardo Mena, por la paciencia que ha tenido conmigo durante todo su transcurso y todo el conocimiento que ha aportado.

También agradecer a Elena Atrián e Isabel Franco, por dejar en manos de un novel como yo su proyecto y poder trabajar sobre él.

En cuanto a los agradecimientos fuera del ámbito de este trabajo, agradecer a mis familiares, amigos y personas más cercanos por haberme acompañado en este camino.

BACTERFIELD: Un videojuego de estrategia para controlar el micromundo

Resumen

En este Trabajo de Fin de Grado se ha realizado la adaptación de un juego de mesa ya existente previamente, llamado Bacterfield, al medio web, para permitir jugar a él de forma online.

El trabajo realizado ha consistido en, primer lugar, analizar e idear como adaptar el juego de mesa, aparte de escoger los lenguajes de programación y herramientas necesarias para su desarrollo

Después, una vez escogidos HTML, CSS y JavaScript para realizar este proyecto, se trabaja en adaptar el funcionamiento del juego de mesa a una página html desarrollando varios prototipos, para garantizar que las mecánicas del juego funcionan en ese entorno. También se ha empezado a trabajar en crear jugadores controlados por la máquina que actúan por ellos mismos sin necesitar interacción del usuario.

Al finalizar estos prototipos se ha conseguido verificar el funcionamiento del juego a nivel local, y se pasa al desarrollo del juego multijugador online, para que pueda funcionar de forma remota. Al terminar este proceso, se ha finalizado el proyecto implementando el uso de IAs en el juego online, y puliendo un poco el diseño de la web.

INDICE

INTRODUCCIÓN	6
1.- ANÁLISIS Y ESTUDIO DEL PROYECTO	10
2.- ELABORACIÓN JUEGO LOCAL	11
2.1.- Cómo crear tablero	12
2.2.- Distintas fases de la partida	14
3.- MEJORAS: CREACIÓN JUEGO ONLINE	16
3.1.- Elección de protocolo	16
3.2.- Implementación	17
4.- RESTO DE MEJORAS AL JUEGO ORIGINAL	23
4.1.- Implementación jugador controlado por la máquina	23
4.2.- Cambios en el tablero original	25
4.3.- Partidas en tiempo real	29
5.- DISEÑO PÁGINA WEB	30
5.1.- Diseño pantallas de la página	30
5.2.- Diseño pantalla de juego	33
6.- CONCLUSIONES	36
6.1.- Desarrollo del trabajo e incidencias	37
6.2.- Posibles ampliaciones al trabajo	41
6.3.- Experiencia adquirida y opinión personal	42
BIBLIOGRAFÍA	44
ANEXO I – MANUAL DE USUARIO	45
I.1.- Página principal	45
I.2.- Unirse a una partida	46
I.3.- Comenzar nueva partida	46
ANEXO II – PUESTA EN MARCHA DE SERVIDOR Y PÁGINA WEB	53
ANEXO III – SISTEMA DE MENSAJES SERVIDOR-CLIENTE	55
III.1.- Mensajes de control	55
III.2.- Mensajes de gestión de partida	57
ANEXO IV – LIBRO DE INSTRUCCIONES JUEGO DE MESA	62

INTRODUCCIÓN

En este proyecto se desarrollará un videojuego basado en BacterField [5], un juego de mesa de estrategia, de dos a cuatro jugadores, inspirado en juegos de estrategia por turnos como Catán o Risk. Los jugadores tendrán que conquistar el tablero colonizándolo con bacterias para cumplir sus objetivos, atacando las bacterias rivales y defendiendo las propias. Para ganar la partida se podrán utilizar diferentes estrategias basadas en procesos biológicos reales, como la producción de antibióticos.

Este juego surge de un proyecto de divulgación [1] que tiene como objetivo dar a conocer a un público no especializado aspectos básicos de la microbiología. Este proyecto de divulgación está liderado por Elena Atrián e Isabel Franco, investigadoras del Instituto de Nanociencia y Materiales de Aragón (INMA-CSIC/UNIZAR), creadoras del juego de mesa BacterField (figura 1) y que aportarán el conocimiento y material gráfico necesario para el correcto desarrollo de este proyecto. El objetivo del TFG será el desarrollo del videojuego para una plataforma online y para el que se desarrollará un algoritmo para poder jugar contra la máquina sin necesidad de más jugadores humanos. También se permitirá un modo de juego en tiempo real, no basado en turnos.



Fig. 1. Juego de mesa en el que se basará este trabajo

El funcionamiento del juego de mesa (Fig. 1), de forma resumida, es el siguiente. Disponemos de un tablero con casillas hexagonales, y cada jugador tiene una celda de salida desde la que empezará con una ficha de bacteria. El objetivo será intentar expandir esa bacteria para llenar el tablero, o intentar frenar la expansión de las bacterias de los rivales. El juego se juega por turnos, y cada turno consta de diversas fases.

En la primera fase, si hemos conseguido generar bacterias con genes de antibióticos, éstas eliminarán bacterias rivales que tengan al lado. En la segunda fase, podremos usar una carta especial (si disponemos de ellas). Estas cartas especiales sirven para otorgar genes a nuestras bacterias, y estos genes servirán para defender a nuestras bacterias de ataques externos, o para hacer que ellas sean las que ataquen a las demás (gen de antibiótico). Una vez usada esa carta, se nos otorga una elección en la siguiente fase. Podremos expandir nuestras bacterias, o podremos usar una carta de acción. Si elegimos expandir las bacterias, podremos generar una nueva bacteria a partir de cada una de las bacterias que tengamos en el tablero, y si esa bacteria tenía algún gen, cuando se expanda lo hará con ese gen también. Si no queremos expandir nuestras bacterias, podemos usar una carta de acción. Existen diversas cartas de acción con distintas funcionalidades, algunas sirven para destruir bacterias enemigas, otras para desplazar nuestras bacterias a cualquier parte del tablero o también algunas se encargan de infectar a bacterias rivales con virus, cosa que dificultará su expansión por el tablero. Ya terminada esta fase, se robará una carta de la baraja y se terminará el turno, pasando a ser el siguiente jugador el que deba realizar estas fases. El juego terminará cuando un jugador alcance el objetivo que se le reparte al comienzo de la partida. Hay varios objetivos, pero la principal diferencia entre ellos es que unos consisten en intentar conquistar el mayor número de casillas del tablero posibles, y otros consisten en intentar eliminar al mayor número de bacterias rivales posible. En el anexo IV viene el libro de instrucciones del juego de mesa original, allí se podrá aprender con más detalle el funcionamiento de una partida.

El objetivo era permitir jugar a este juego desde cualquier navegador Web de forma remota, porque de esta forma facilitamos el acceso al juego y evitamos acciones tediosas como tener que descargar e instalar el juego, así que se usarán los siguientes lenguajes informáticos: HTML, CSS y JavaScript.

Otro aspecto importante de este TFG, aparte de replicar el funcionamiento y mecánicas del juego de mesa original en un entorno Web, es intentar aportarle valor ampliando las fronteras

que le impone el medio físico, e introducirle diversas mejoras al juego que estarán disponibles en el juego online. Las mejoras de las que se hablan serán:

- Permitir partidas con más jugadores. El juego original estaba limitado de 2 a 4 jugadores, pero aquí se permitirán partidas de hasta 10 jugadores, y para lograr eso se alterará el tamaño del tablero y la distribución de casillas de forma dinámica y automática, intentando que el juego siga teniendo un equilibrio.
- Diseñar jugadores controlados por la máquina: Aparte de poder jugar contra otros jugadores de forma remota, permitir jugar contra jugadores controlados por la máquina, con la elaboración de un algoritmo para jugar contra la máquina teniendo ésta diversos comportamientos, aunque el profundizar en estas técnicas no es el objetivo principal del proyecto, por lo que su comportamiento será sencillo pero suficiente para cumplir su función. Los scripts de estos jugadores les permiten realizar las mismas acciones que haría un ser humano, pero de forma automática, y además están parametrizados para que actúen de forma distinta según las preferencias del usuario.
- Modo de juego en tiempo real: El juego original es por turnos, cuyo funcionamiento se ha explicado anteriormente, y aunque en este proyecto la idea inicial era introducir esa forma de juego, también se añadirá una forma de juego en la que no funcione por turnos, sino que será a tiempo real, y todos los jugadores estarán realizando acciones a la vez sin esperar a que vuelva a ser su turno o a que el resto de jugadores hayan terminado sus elecciones.
- Partidas online: Permitir jugadores online, ya sea creando partidas o uniéndose a ellas. Para la comunicación entre servidor y jugadores, se usarán las librerías node.js y socket.io, que sincronizarán los mensajes de todos los jugadores y actualizarán todos los tableros conforme los jugadores realicen acciones en el suyo propio. En las partidas online podrá haber distintos tipos de jugadores. Los jugadores podrán ser jugadores humanos o jugadores controlados por una IA, además, los jugadores pueden ejecutarse de forma local o pueden conectarse de forma remota al juego, jugando online. Da igual si el jugador juega de forma local o remota, podrá elegir ser un jugador controlado por un humano o controlado por la máquina.

Las secciones que compondrán esta memoria son las siguientes. En primer capítulo se hablará del análisis del proyecto y elección de herramientas a usar. En el segundo capítulo se hablará de la elaboración juego local, dónde se explicará la elaboración de un prototipo a nivel

local fiel al juego original de tablero y a sus normas. El tercer capítulo tratará sobre la creación del juego online, ahí se describirá como se ha pasado de los prototipos a nivel local iniciales, a crear la estructura que permitirá el funcionamiento multijugador remoto. Después, en el siguiente capítulo, se hablará de las mejoras al juego original, cuyo resumen es describir como se crea la IA y cómo funciona, a diferencia de cómo funcionan los jugadores controlados por humanos. También se explicarán algunas de las mejoras introducidas al juego original. Por último, en el capítulo final, se mostrará el diseño de la página Web, explicando cómo se han diseñado los diferentes apartados de la Web.

1.- ANÁLISIS Y ESTUDIO DEL PROYECTO

En este apartado se explicará como comenzó este TFG y los estudios y evaluaciones que se hicieron en ese momento.

La elección de este proyecto surgió por la motivación de aprender a crear entornos Web en los que además exista una comunicación simultánea entre varios clientes. La adaptación de un juego de mesa parecía una buena forma de lograr esto. Lo primero fue plantearse que elementos se deberían elaborar para poder completar el proyecto.

El objetivo del trabajo es lograr que este juego se pudiera jugar desde cualquier navegador, y para ello la forma más sencilla de desarrollar nuestro entorno Web es el uso de los muy comunes lenguajes informáticos conocidos como HTML y CSS, que servirían para el diseño de la Web y sus distintos apartados. Después, se tuvo que escoger que lenguaje usar para la programación del propio juego en sí. Quizás en anteriores años se podrían haber tenido más alternativas para el desarrollo Web, pero con el fin de soporte de Flash Player a finales de 2020, estas alternativas se redujeron. Quizás una forma sencilla hubiera sido hacer el juego en plataformas especializadas para el desarrollo de juegos como Unity, pero se encontró que tendría más valor programar puramente en la Web, y por ello se optó por usar JavaScript.

Una de las ventajas de usar JavaScript es que está orientado al diseño de aplicaciones en navegadores Web. De hecho, un 97.6% de las páginas Web hacen uso de JavaScript como programa para desarrollo de aplicación desde el lado del cliente [2]. Luego, al igual que HTML y CSS, no es un lenguaje que hayamos aprendido en este grado, pero que sí posee características de programación que hemos estudiado en diversas asignaturas, y que se podrían aplicar aquí, como la orientación a objetos. Aparte, comparte características con lenguajes de programación como C o Java con los que sí se ha trabajado en este grado.

Así que simplemente con el uso de esos tres lenguajes, y con el material gráfico e intelectual que fue aportado por las creadoras del juego de mesa original, ya se poseían los elementos suficientes para el desarrollo de todo este proyecto, por lo que el paso inicial fue el estudio y aprendizaje de estos lenguajes, ya que era un conocimiento que no se tenía al empezar este trabajo, y mi experiencia en el desarrollo Web era nula, por lo que se partía de cero.

2.- ELABORACIÓN JUEGO LOCAL

En este apartado se explicará cómo se idearon y realizaron los primeros prototipos del juego que funcionarían a nivel local.

Antes de empezar a realizar los prototipos del juego en sí, se estudiaron los lenguajes HTML, CSS y JavaScript, y se realizaron y estudiaron pequeñas pruebas y demos de otro tipo de juegos para aprender a programar.

Una vez concluidos esos estudios iniciales, se empezó a desarrollar el prototipo del juego. Este prototipo tenía como objetivo replicar el funcionamiento del juego de mesa original, es decir, permitir poder jugar una partida con cuatro jugadores, en la que se pudieran expandir las bacterias, usar las cartas y todas las acciones que aparecen en un turno de la partida, todo esto repitiéndose para cuatro jugadores. Al ser un prototipo, simplemente se quería verificar el funcionamiento básico del juego, no se incidió mucho en el apartado gráfico de la Web, y solo se creó la página HTML que incluía el juego. Además, los cuatro jugadores jugarían desde la misma pestaña del navegador Web, algo que no es concebible en el producto final, pero que para ser usado como punto de partida era más que suficiente.

El primer elemento a realizar en el prototipo era dibujar el tablero (Fig. 2) en la página Web. Se podría haber intentado alterar el tablero o la forma en la que se veía respecto al juego de mesa original, pero se decidió ser fiel a él e intentar replicarlo lo máximo posible. Con esto lo que se quiere decir es que el tablero se vería en dos dimensiones, visto desde arriba cómo vemos un tablero en la vida real.



Fig. 2. Tablero del juego de mesa original

Tras elegir eso, lo siguiente fue escoger cómo se implementaría el tablero en la página Web. La opción sencilla sería realizar una imagen del tablero en cualquier programa de edición o dibujo, copiarlo y pegarlo en la página, y ya colocar las fichas encima de esa imagen. Y aunque esa opción fue con la que se empezó el prototipo, se vio que no dejaba mucha flexibilidad al desarrollo del proyecto, ya que se estaba limitando el juego a la forma de jugar del juego original, solo hasta cuatro jugadores. Así que, para intentar buscar más flexibilidad de cara al futuro del trabajo, se buscó una forma dinámica de dibujar la página Web. La opción escogida fue canvas, elemento de HTML que permite dibujar gráficos usando JavaScript, a través de comandos y funciones.

2.1.- Cómo crear tablero

Para dibujar el tablero con casillas en forma hexagonal, se buscó información sobre cómo hacerlo, y se encontró una página Web (redblobgames) [3] de gran utilidad que realizaba un exhaustivo trabajo en cómo implementar tableros hexagonales, ya fuera tanto para dibujarlo, cómo para también colocar elementos en las casillas que quisiéramos o localizar sobre que casillas interactúa el usuario, además de más funcionalidades.

Siguiendo las explicaciones y código proporcionado por esa página, se realizó el dibujo del tablero. Hubo que realizar distintas elecciones para crear el tablero e intentar imitar el estilo original. El primero era la forma de las casillas, que eran hexágonos, en concreto hexágonos “flat-topped” (Fig. 3), que lo que quiere decir es que la cima del polígono es un lado del hexágono, no una arista.

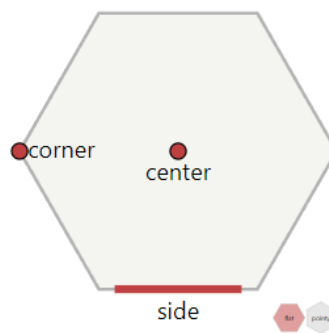


Fig. 3. Hexágono “flat-topped” proveniente de [2]

Lo siguiente es estudiar las dimensiones del hexágono (Fig. 4), para luego ser capaz de usarlas para dibujarlos en el píxel que deseemos. Se llama tamaño a la distancia del centro a cualquier arista. El ancho del hexágono es el doble de ese tamaño, y la altura es ese tamaño multiplicado por la raíz cuadrada de 3 (seno de 60°). Con esto podemos averiguar la distancia entre hexágonos adyacentes, que horizontalmente será el ancho del hexágono multiplicado por 0,75, y verticalmente será la altura del hexágono 2. Con estos elementos ya podemos dibujar un hexágono en el canvas y poder colocarlos uno detrás de otro.

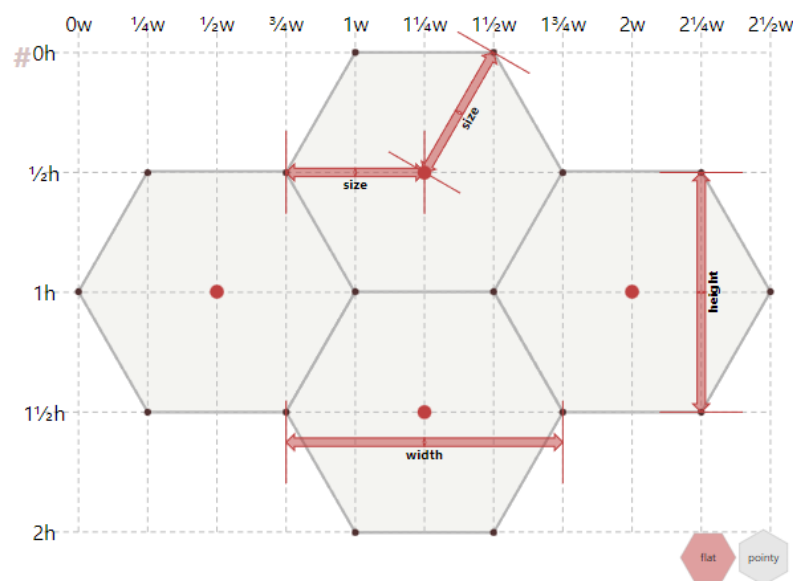


Fig. 4. Dimensiones del hexágono, proveniente de [2]

Una vez dibujado el tablero en el canvas, lo siguiente que necesitamos para implementar nuestro juego es implementar un sistema de coordenadas, para localizar dónde está cada casilla y reconocer los movimientos que realizan los jugadores. Existen varias alternativas explicadas en el documento anteriormente nombrado, pero se usará un sistema simple consistente en coordenadas basadas en filas y columnas (Fig. 5). El índice de las filas viene indicado por la posición horizontal de la casilla, todas a la misma altura, y el de las columnas va creciendo con la casilla adyacente en diagonal hacia abajo y la derecha.

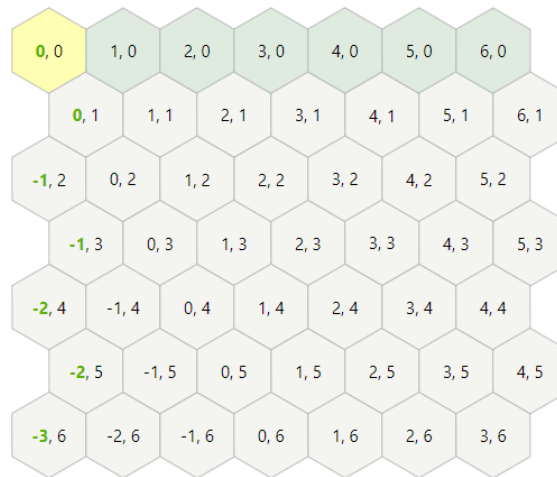


Fig. 5. Cómo se generan las coordenadas del tablero, proveniente de [2]

2.2.- Distintas fases de la partida

Sabiendo la coordenada de una casilla, podemos conocer en qué localización estará en la página, hallando las coordenadas en píxeles en las que está, así que podremos colocar los elementos que queramos allí. También sabremos el camino inverso, es decir, si un usuario clicla en una casilla, con los píxeles de la página que recojamos del evento del click, podremos relacionarlos con una coordenada del tablero. Esto es vital para el siguiente paso, que es una vez dibujado el tablero, debemos colocar en él las fichas del juego, que son las de bacterias (con sus distintos tipos) y las de biofilm. Las fichas serán imágenes que tendremos guardadas, y a las que el juego accederá y colocará en la casilla que desee el jugador. El jugador escogerá la casilla clicando sobre ella.

Otra situación del turno es la expansión de bacterias, para lo que se recopilan todas las bacterias del jugador del turno que hay, marcando su fondo en rojo para que el jugador sepa cuales son. Luego el jugador clicla en ella y se iluminan las casillas vecinas, y el jugador escoge a cuál se expande. El programa puede reconocer las casillas vecinas gracias a las coordenadas que acabamos de explicar.

Una vez implementado el tablero y la posibilidad de colocar elementos en él, lo siguiente es implementar el uso de las cartas. En el prototipo las cartas primero se probaron poniéndolas en texto, y que al cliclar el usuario en ella pudiera realizar la reacción que quisiera. El siguiente

paso ya fue habilitar un espacio en la web en la que se recogieran las cartas del jugador con sus imágenes, y que él clicara en ellas para usarlas.

El último paso de este prototipo, fue implementar los turnos y las distintas acciones que se realizarán en ellos, que son equivalentes a las del juego de mesa. Se añade un cuadro con botones que clicar para avanzar de fase del turno, así hasta el turno siguiente. Se recuerda que este prototipo estaba hecho a nivel local, sin funcionar en Internet, y todo sucediendo en la misma pestaña. Así que, una vez terminado un turno de un jugador, se pasa al siguiente eliminando las cartas del anterior jugador y añadiendo las del siguiente. Es cierto que en el juego de mesa no se debería saber las cartas que tienen el resto de jugadores, pero esto era un prototipo así que solo se quería saber si el juego funcionaba o no.

3.- MEJORAS: CREACIÓN JUEGO ONLINE

Con el juego ya funcionando en su primera fase, a nivel local, lo siguiente es conseguir que se pueda jugar a él de forma remota entre varias personas

3.1.- Elección de protocolo

Nos encontramos con la tesitura de adaptar un juego de mesa, así que al no ser un juego muy complejo o que vaya a requerir muy poca latencia para garantizar una experiencia de usuario óptima, se opta por usar conexiones TCP entre usuarios y servidor. Tras eso, se debe decidir que protocolo de actuación se va a seguir para el funcionamiento de la partida y la interacción entre clientes y servidor. Después de buscar información, se elige el protocolo WebSocket para que cliente y servidor se comuniquen.

WebSocket nos proporciona comunicación full-duplex sobre una conexión TCP, así que cliente y servidor se pueden estar mandando mensajes de forma simultánea (Fig. 6). De esta forma el servidor puede enviar mensajes al cliente cuando quiera, sin necesidad de que el cliente se los solicite, y así, por ejemplo, si un cliente quiere realizar una acción, el servidor puede comunicársela al resto de clientes de forma inmediata, para que todos se sitúen en el mismo estado de la partida a la vez. Esto va a hacer que la comunicación sea muy fluida.

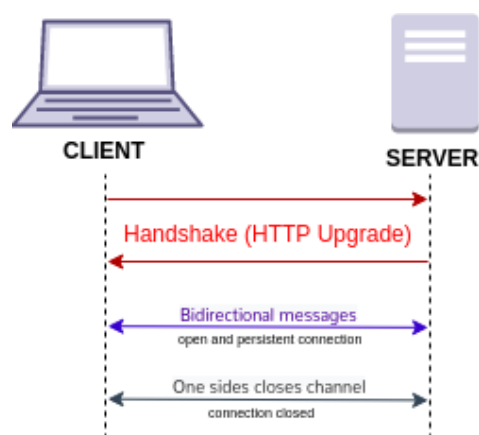


Fig. 6. Funcionamiento conexión con WebSocket. Imagen obtenida de <https://commons.wikimedia.org>

3.2.- Implementación

Para usar WebSocket, necesitaremos las siguientes librerías de JavaScript [4]:

- Node.js
- Express.js
- Socket.io

Node.js y Express.js sirven para poner en marcha el servidor y vincularle los scripts que tendrá que ejecutar, pero la más importante en este caso es socket.io, ya que ella nos proporciona una forma simple de intercambiar mensajes entre cliente y servidor.

En la parte del servidor, socket.io se encarga de recibir las nuevas conexiones. Proporciona la capacidad de separar los nuevos sockets que se forman en distintas salas, algo que será muy útil para nuestro juego, ya que destinaremos a todos los jugadores de una partida a una misma sala. Para hacer esto, solicitaremos que el usuario que cree la partida proporcione una contraseña, y crearemos una sala en el servidor asignada a esa contraseña. Así, cuando lleguen nuevos usuarios que introduzcan esa contraseña, se les introducirá en esa sala creada para esa partida. Se podría implementar una opción para empezar la partida cuando se llene la sala, pero se ha optado por dejar esa elección a la persona que ha creado la partida, así que cuando ella quiera, inicia la partida mandando un mensaje al servidor, y este envía una alerta a los clientes para que empiecen a jugar.

Después de estar creada la partida, clientes y servidores se van a comunicar para actualizar el tablero. El cliente realiza sus acciones en el turno, y si están envuelven algún cambio en el tablero o partida, mandan un mensaje para notificar al servidor. El servidor nada más recibir este mensaje, mandará el mensaje oportuno a los clientes que están conectados en la misma sala de juego, y así de esta forma todos actualizan su estado a la vez. De esta forma es como si todos estuvieran jugando sobre el mismo tablero de forma simultánea, aunque estén conectados de forma remota, emulando al juego de mesa.

Existen mensajes para repartir las cartas a los clientes y para que éstas vuelvan al mazo, ya que a todos los efectos el mazo de cartas de la partida está almacenado en el servidor. Otros mensajes colocan y destruyen las fichas del tablero cuando un cliente realiza esa acción en su tablero. Luego existen mensajes que no tienen que ver puramente con el desarrollo de la partida,

sino que funcionan como mensajes de control (Fig. 7). Unos nos sirven para que un cliente pueda unirse a la partida, recibiendo todos los datos de la partida para establecer su tablero, otros nos sirven para gestionar desconexiones, permitiendo así que cuando un usuario se desconecta, él mismo u otro pueda unirse a la partida, y parta desde la situación en la que se quedó el usuario cuando se desconectó, recibiendo el estado en el que se encuentra el tablero o las estadísticas que llevaba el usuario previamente a su desconexión. Finalmente existe un mensaje para finalizar la partida que se manda cuando un usuario consigue su objetivo y gana la partida.

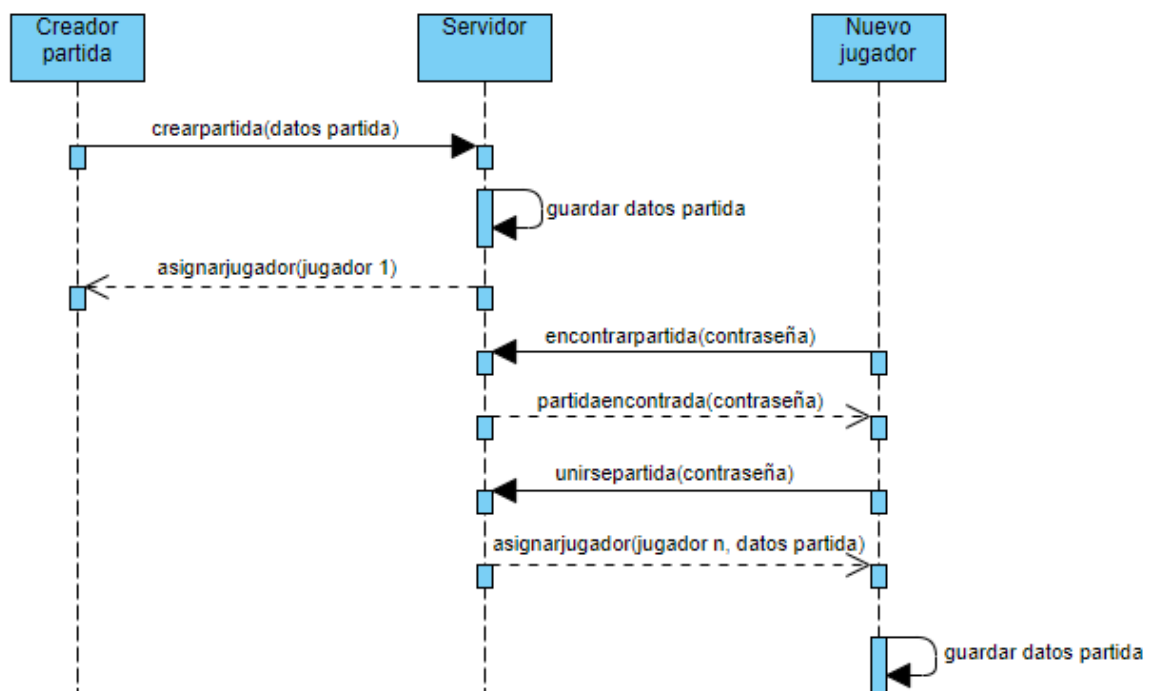


Fig. 7. Diagrama de secuencia representando la creación y establecimiento de una nueva partida

Con el anterior diagrama (Fig. 7) se muestra cómo se crea la partida, pero ahora se va a explicar cómo funciona el transcurso de la partida, poniendo como ejemplo los mensajes que se transmiten en el turno de un jugador. Lo primero será ponernos en situación, mostrando el estado de la partida (Fig. 8).

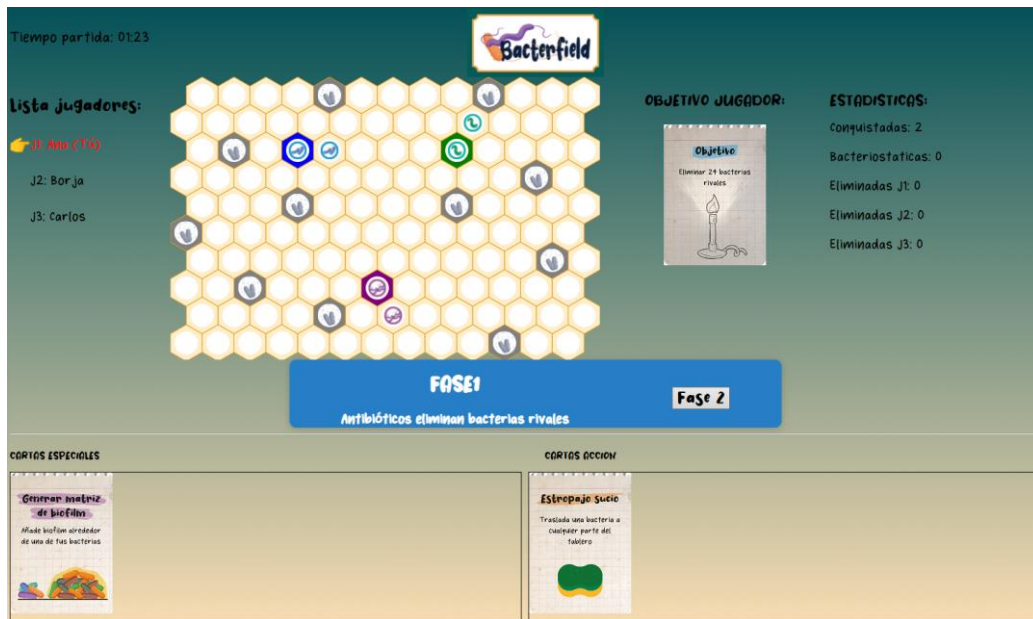


Fig. 8. Pantalla que ve el jugador 1 al inicio del turno

Somos el jugador uno (fichas de color azul) y es nuestro turno. Como aún no tenemos fichas con genes de antibiótico, en esta fase no sucede nada. Lo siguiente que haremos será pasar a la fase 2, en la que podremos usar una carta especial si queremos. Como tenemos una carta de generar biofilm en la mano, la usaremos con una de nuestras fichas, en concreto la que está en la coordenada (3,2). El paso de mensajes que se realiza se ve en la Fig. 9. Después de esos mensajes, se actualiza el tablero de todos los jugadores y se ve como en la Fig.10.

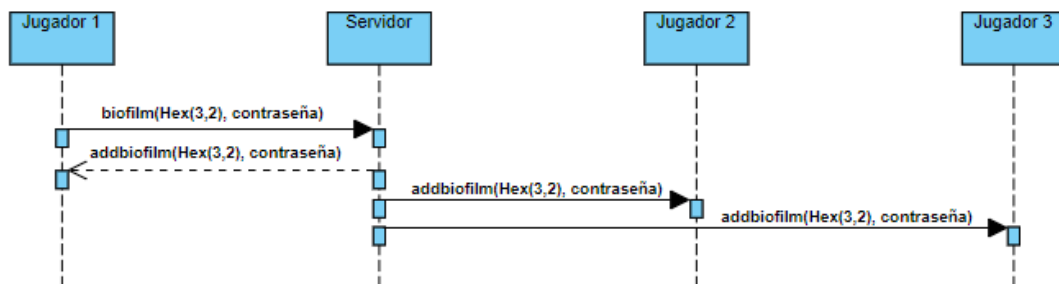


Fig. 9. Diagrama de secuencia representando como se añaden las fichas de biofilm



Fig. 10. Tablero del jugador 1 una vez usada la carta de biofilm

Ahora podemos escoger entre usar la carta de acción (estropajo) o expandir nuestras bacterias. Como tenemos pocas fichas en el tablero, escogemos expandir bacterias. Clicamos sobre la bacteria que queremos expandir, y luego sobre la celda que queremos ocupar. Hacemos eso con las dos bacterias, añadiendo así dos nuevas bacterias de nuestro tipo al tablero. En la Fig. 11 vemos el paso de mensaje que garantiza que a todos los jugadores se les actualice el tablero, y en la Fig. 12 vemos como queda el tablero después de enviar y recibir esos mensajes.

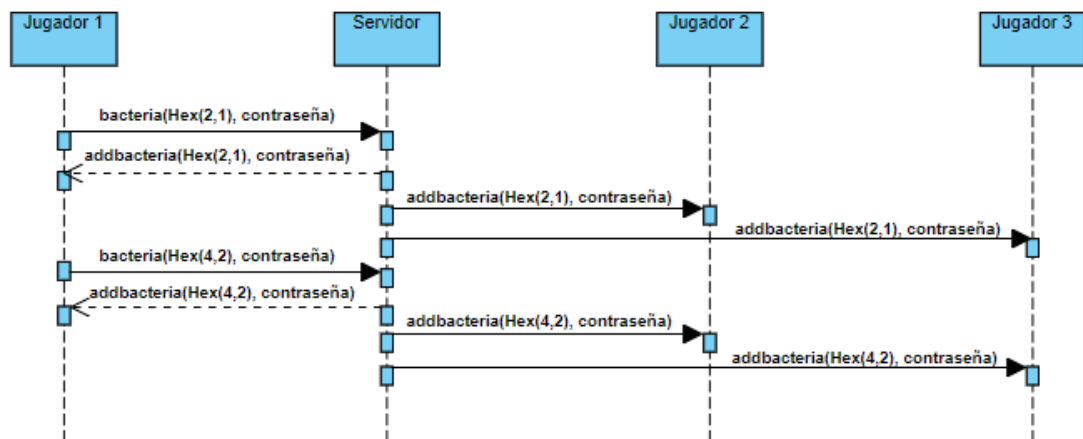


Fig. 11. Diagrama de secuencia representando como se añaden las bacterias al tablero



Fig. 12. Tablero del jugador 1 después de expandir sus bacterias

Después de expandir las bacterias, ya llega la hora de terminar el turno. Cuando lo solicitemos, robaremos una carta de la baraja (el servidor nos la enviará) y le indicaremos al servidor que se acaba el turno, para que él se lo comunique a los demás. En la Fig. 13 veremos el diagrama de ese paso de mensajes, y en la Fig. 14 veremos cómo queda el tablero una vez robada la carta y pasado el turno.

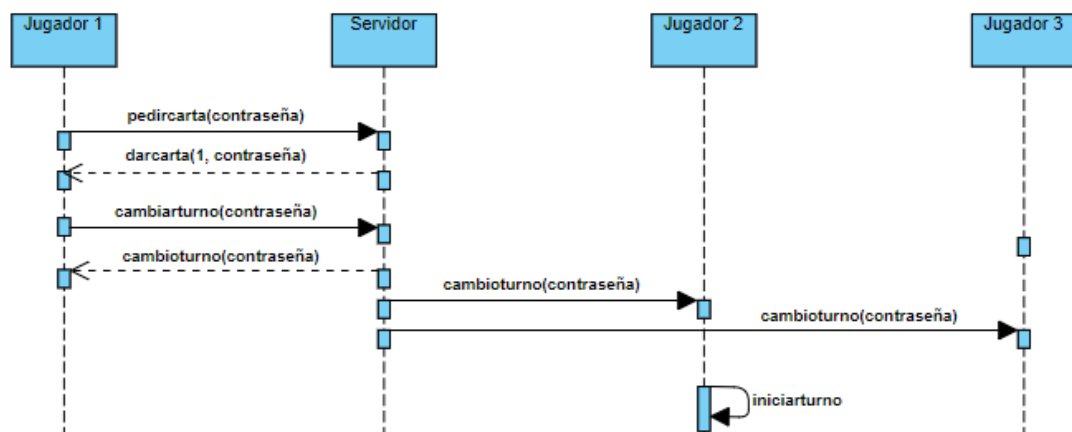


Fig. 13. Diagrama de secuencia representando el final del turno del jugador 1



Fig. 14. Pantalla del jugador 1 después de pasar de turno

Con todos estos tipos de mensajes, gestionamos el juego y garantizamos que se pueda jugar una partida de forma remota, de tal forma que se juegue como si estuviéramos en local. En el anexo III se listarán todos los tipos de mensajes que forman parte de este sistema de comunicación entre servidor y jugadores.

Por el momento, estamos hablando de partidas por turnos, lo que quiere decir que hay que esperar a que un jugador pase el testigo al siguiente para que se cambie el turno. ¿Pero qué sucede cuando un jugador se ha desconectado? Hay un hueco ahí en el que no hay ningún jugador que sea capaz de pasar el testigo de cambio de turno al siguiente. Lo que se hace en este caso, es saltarse a ese jugador, y que no realice ninguna acción. Para poder hacer esto, el resto de jugadores deben conocer que jugadores están desconectados de la partida, dato que les llegará a través del servidor cuando éste se dé cuenta de que un jugador se ha caído de la partida.

4.- RESTO DE MEJORAS AL JUEGO ORIGINAL

Uno de los propósitos de este trabajo, aparte de replicar el juego de mesa en un entorno online, era ampliar las fronteras impuestas por el tablero, y añadir nuevas funcionalidades. A continuación, pasamos a explicar algunas de ellas.

4.1.- Implementación jugador controlado por la máquina

Una mejora al juego de mesa original, es que aprovechando que nos encontramos en un entorno virtual, se pueden implementar partidas en las que algunos jugadores no sean controlados por humanos, sino que sean controlados de forma automática por el programa.

La primera implementación del juego, que solo estaba hecha para ser controlada por personas, confiaba mucho en la interacción entre usuario y página para realizar las distintas acciones de la partida. Por ejemplo, el usuario escoge que carta usar clicando en ella, o escoge que bacteria mover o expandir clicando en la celda que desee. Así que, para poder usar un jugador controlado por la máquina, se deben implementar procesos en el programa para recopilar el estado del tablero y todas las celdas, y una vez obtenidas todas nuestras bacterias y fichas, el programa puede escoger cual usar o expandir. En principio lo hace de forma aleatoria, seleccionando un número al azar, pero se decidió implementar algoritmos con distintos comportamientos.

El conocimiento sobre inteligencia artificial era bastante básico, así que, para añadir distintos comportamientos, lo que se hace es aportar distintas probabilidades a las decisiones que debe tomar el programa, dependiendo del tipo de jugador que se escoge. Las decisiones que tomará el algoritmo sucederán cuando en el juego normal el usuario escogía cambiar de fase en el turno (como podemos ver en la Fig. 15). Además, de los objetivos que tenemos en el juego, unos consisten en conquistar celdas y otros en destruir bacterias, así que dividimos el comportamiento del algoritmo en conquistador de celdas o destructor de bacterias en función de su objetivo.

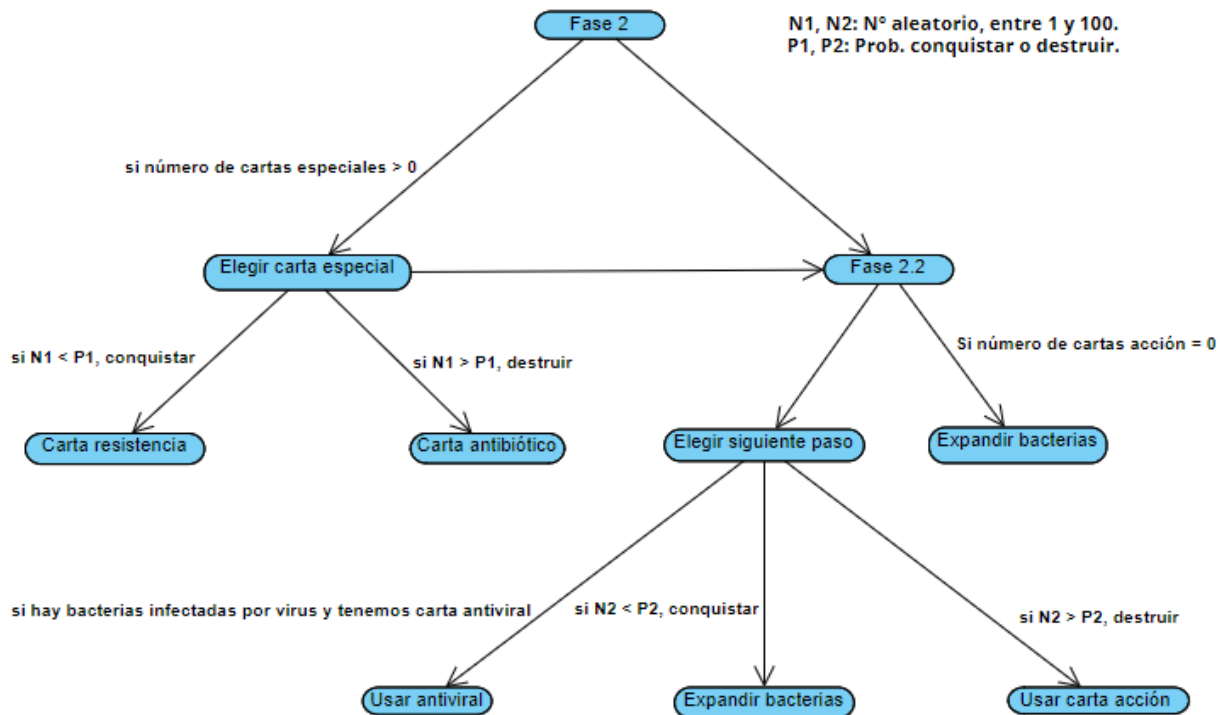


Fig. 15. Elecciones que hace el algoritmo en un turno

Cuando tenga que escoger entre una carta que destruya bacterias o si quiere expandir sus bacterias en el turno, lo hará dependiendo de su objetivo.

Un algoritmo hará eso de forma fiel, sin contemplaciones, y otro tendrá un comportamiento un poco distinto, ya que la probabilidad de realizar una acción según su objetivo no será del 100%, sino un porcentaje mejor para admitir errores en el comportamiento del algoritmo, haciéndolo más real.

Así, los tres tipos de comportamiento que tenemos es:

- IA1: Todas las elecciones las hace al azar, 50% de probabilidades.
- IA2: Otra que intenta siempre avanzar hacia su objetivo, 100% probabilidad dependiendo de si quiere conquistar o destruir. Si quiere conquistar, a la hora de usar una carta especial tendrá preferencia por usar una carta de resistencia o de biofilm, para facilitar su crecimiento, y cuando tenga que decidir entre expandir bacterias o usar una carta de acción, elegirá expandirse. Al contrario, si quiere destruir, dará prioridad a usar la carta especial de antibiótico, que elimina a las bacterias rivales adyacentes, y preferirá usar una carta de acción que destruya

- IA3: La última, como la anterior, pero bajando la probabilidad (en el prototipo se ha bajado al 70%), para que la IA parezca más un jugador real, admitiendo errores en su estrategia.

En la Fig. 15 se ve por encima como realiza las elecciones el algoritmo. Allí aparecen las decisiones principales que hace, que es entre destruir o conquistar. Pero hay que decir que, a la hora de elegir cartas, hay más probabilidades introducidas para hacer la elección, porque dependerá también de las cartas que tengamos en la mano para hacer la elección, así que hay una probabilidad para cada posible combinación de cartas que tengamos.

En el capítulo de conclusiones de este trabajo, se propondrán distintas mejoras posibles a los algoritmos que dirigen los comportamientos de los jugadores controlados por la máquina.

4.2.- Cambios en el tablero original

El tablero original está pensando para partidas de dos a cuatro jugadores, y tiene un tamaño fijo, como es lógico en los juegos de mesa. Así que otra mejora introducida en el juego es permitir partidas de más de cuatro jugadores.

Para esas partidas de más de cuatro jugadores, se amplía el tamaño con cada jugador introducido. Así, podrían añadirse todos los jugadores que se quisieran a una partida, pero se decide añadir un límite de 10 jugadores para que los tableros no sean exageradamente grandes, aunque es algo que podría cambiarse.

Además, cabe mencionar que la distribución de las celdas de salida y la de las casillas bacteriostáticas (casillas grises como nos referiremos a ellas a partir de ahora) se realiza de forma automática mediante un algoritmo, lo que quiere decir que este programa no cuenta con una cantidad x de tableros preestablecidos, sino que se van creando en tiempo de ejecución del programa. Es cierto que lo más sencillo hubiera sido crear una cantidad fija de tableros y que el usuario escogiera entre ellos, pero eso restringía bastante las opciones que se le daban, y además impedía que, si en un futuro se deseaba que este juego pudiera ser jugado por más jugadores que los del límite establecido, ello se pudiera implementar de forma fácil en el programa, ya que habría que haber creado a mano los nuevos tableros, rellenando a mano todas sus filas y columnas. Esto proporciona al programa gran versatilidad y capacidad de

crecimiento, porque simplemente con cambiar el valor del límite de jugadores en partida, el usuario ya podrá escoger más jugadores y se crearán tableros para satisfacer sus necesidades.

El número de casillas grises que hay en el tablero depende del número de jugadores en partida. Como en el juego original hay cuatro casillas de este tipo por cada jugador, se decide respetar esta proporción y usarla para distribuirlas en el tablero. Ya sabiendo el número de casillas grises, ahora se deberá distribuirlas por el tablero. Lo primero es mirar el número de filas que hay. Tendremos un vector de longitud igual al número de filas, y en él se indicará cuantas casillas grises habrá en cada fila. Así, si el número de casillas grises a colocar es menor al número de filas, lo que se hará es distribuir esas casillas de forma proporcional en ese vector. Ejemplo:

-Tenemos 11 filas y 8 casillas grises que colocar. Lo que hará nuestro programa será devolvernos un vector con este aspecto: [1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 1]. Los unos quieren decir que en esa fila habrá una casilla gris, y los 0 que en esa habrá cero casillas grises.

En el caso de que las casillas grises a colocar fueran superiores a las filas, se sumaría una casilla a todas las filas, y luego se volvería a evaluar esto con las casillas grises restantes que colocar. Cuando se llegue a la situación de tener menos casillas grises que filas, se realizará la distribución antes explicada.

Ya sabiendo cuantas casillas grises hay en cada fila, se van colocando en el tablero. Si solo hay una, se coloca en una posición inicial, y a esa posición se le suma un valor en el que se colocará la siguiente gris que esté sola en su fila. Si hay más de una gris en una fila, éstas dos se distribuyen de forma uniforme de forma similar a la que hemos explicado antes, lo único que ahora la longitud del vector sería la cantidad de columnas del tablero. En la Fig. 8. observamos un ejemplo de colocación de casillas grises. En el ejemplo hay tres jugadores, por lo que habrá 12 casillas grises que colocar. El tablero tiene 10 filas, excediendo así el número de casillas grises a las filas, por lo que primero se rellena el vector antes explicado con unos. Eso nos devuelve un vector [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]. Con 10 casillas ya asignadas, solo nos quedan dos, que se distribuirán de forma homogénea por el vector de 10 huecos. Esas dos casillas se colocarán de esta forma: [0, 0, 0, 1, 0, 0, 0, 1, 0, 0], y ese vector se sumará al anterior, con lo que logramos ya el vector definitivo de distribución de casillas grises respecto a las filas: [1, 1, 1, 2, 1, 1, 1, 2, 1, 1]. Podemos ver que esa distribución se ve reflejada en el ejemplo de la Fig. 16.

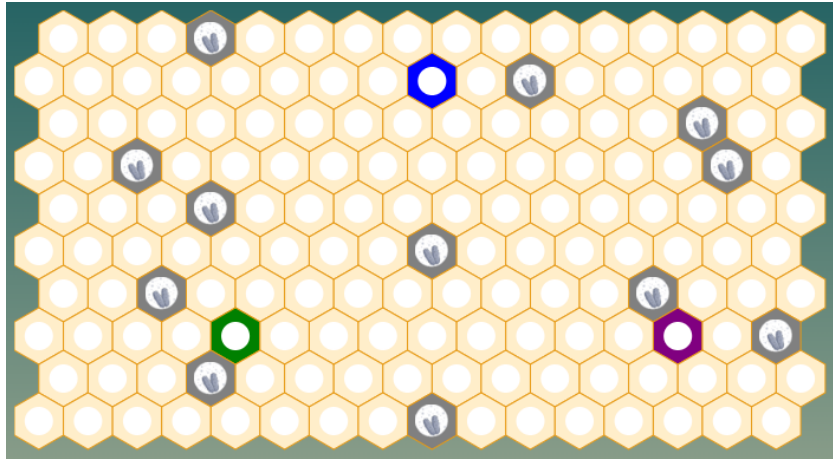


Fig. 16. Ejemplo de cómo se distribuyen las celdas de salida en los óvalos imaginarios

El sistema para colocar las casillas de salida es distinto al de casillas grises. Lo que se quiere conseguir en este caso es que las casillas de salida se encuentren todas a una distancia similar respecto a sus contiguas, y que intenten ocupar el tablero de forma homogénea. Aunque la primera tentativa fue hacer un modelo similar al de las grises, repartiendo las casillas de salida de forma equitativa a través de todas las filas, se optó por otra forma de administrarlo. Lo que se hace es dibujar un óvalo imaginario dentro del tablero (con un radio horizontal y otro vertical) y distribuir las celdas de salida de forma homogénea a través de él. Esto se hace colocando la primera celda de salida a “las 12 en punto”, y luego partiendo ese óvalo en n partes (siendo n el número de celdas que colocar), dividiendo 2π radianes (una vuelta entera) entre n . Además, se ha implementado que llegado a un número de jugadores (el número decidido es 8, pero podría efectuarse a partir del número de jugadores que se quisiera) se añada un óvalo interior más pequeño, dentro del otro que ya habíamos dibujado (Fig. 17). Así hacemos que el centro del tablero no esté muy despoblado y que los jugadores no estén demasiado dispersos.

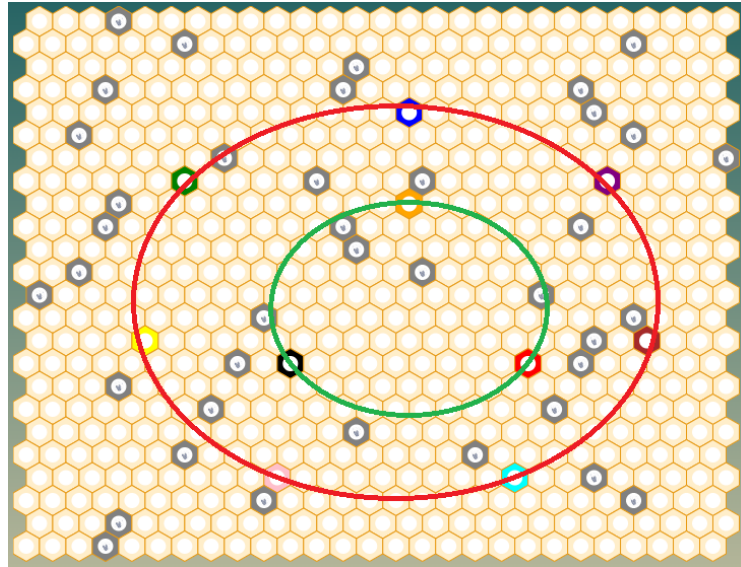


Fig. 17. Ejemplo de cómo se distribuyen las celdas de salida en los óvalos imaginarios

Las coordenadas en las que se ponen se calculan con el radio horizontal del óvalo por el coseno del ángulo que incrementamos (x) y con el radio vertical por el seno de ese mismo ángulo (y). Con esas formas ya podemos deducir que casilla será la de salida, porque hay una función en el programa que traduce una coordenada en píxeles a la casilla a la que corresponde, teniendo en cuenta el tamaño del hexágono escogido y la localización de la casilla de origen, que es la casilla con coordenadas (0,0). Observamos ahora un ejemplo de esos dos sistemas en funcionamiento, en ese caso creando un tablero de 5 jugadores (Fig. 18).

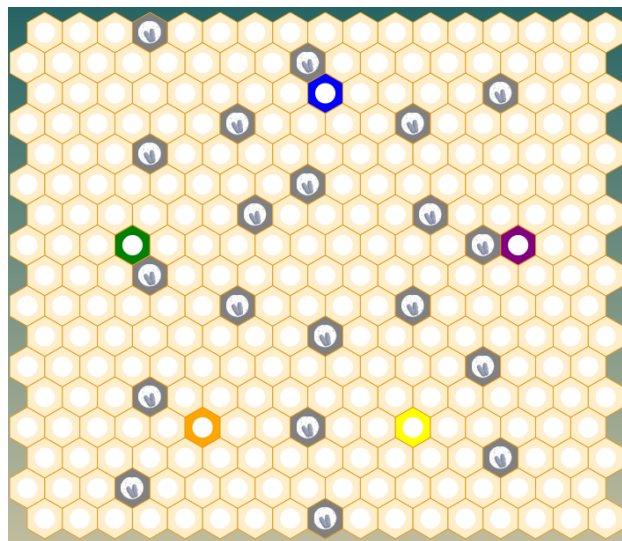


Fig. 18. Ejemplo de tablero creado automáticamente para 5 jugadores

Aparte de esas medidas, en la creación de partida también se dejará al usuario cambiar el ancho y alto del tablero que viene por defecto, pudiendo escoger tableros más grandes en los que jugar.

4.3.- Partidas en tiempo real

La siguiente mejora es añadir otro tipo de partidas, en las que no haya turnos, sino que todos los jugadores estén realizando acciones a la vez. Esto se hace para añadir un modo nuevo de juego, alterando el funcionamiento normal de la partida, para que los jugadores experimenten y tengan más variedad. Cuando sean en tiempo real, podremos jugar como se juega en los juegos de estrategia en tiempo real, sin tener que esperar a que los otros hagan sus acciones. Además, el mundo real de las bacterias se asemeja más a este comportamiento que al de una partida con turnos, aspecto que aporta también al factor de divulgación que tenía este proyecto.

Para implementar esta mejora, lo que se realiza es lo siguiente. En nuestro programa existe una variable denominada “jugador”, que nos dice que jugador somos, y una variable “turno”, que indica el turno en el que nos encontramos actualmente. Normalmente, cuando se terminan todas las fases de un turno, se manda un mensaje al servidor avisando de ello, y solicitando cambiar turno, para que esto les llegue a todos los clientes. Pero en este caso no se hace eso, sino que al final del turno el jugador vuelve a iniciar su turno sin comunicar un cambio de turno. Además, todas las funciones que antes dependían de la variable “turno” para saber qué tipo de fichas poner en el tablero, ahora usarán la variable “jugador”.

Cabe comentar que la elección de WebSocket como protocolo de comunicación es un elemento que ha sido muy útil para implementar esta mejora. Como ya se ha comentado antes, WebSocket permite establecer una conexión entre sockets que permanezca siempre abierta y sea full-duplex, por lo que los clientes pueden estar enviando mensajes al servidor y recibiendo mensajes de él de forma constante, lo que permitirá que él efectúe cambios en el tablero a la vez que un jugador rival efectúe cambios en él. Si se hubiera elegido un sistema de comunicación estándar, en el que el cliente debe solicitar al servidor que le envíe los datos que quiere ver, no habría sido posible implementar partidas sin turnos en las que un jugador recibe actualizaciones constantes del servidor, aunque no las desee.

5.- DISEÑO PÁGINA WEB

Una vez ya implementado todo lo relacionado con el juego, y ya siendo funcional, la última fase de este trabajo es diseñar la página web en la que estará el juego, y determinar la estructura que tendrá desde que el usuario se conecta a la dirección Web, hasta que empiece la partida. Hasta ahora hemos hablado solo de cómo crear el juego, pero ahora habrá que añadir distintas páginas que permitan configurarlo o unirse a él.

Para este apartado, se realizó una fase de diseño en la que se hicieron varios bocetos para barajar el posible aspecto de la página. Hay un espectro muy amplio de decisiones a la hora de diseñar una página web, pero se optó por buscar la simplicidad para que la página fuera lo más accesible posible, y no añadir opciones redundantes o que no aportaran mucho al usuario.

5.1.- Diseño pantallas de la página

El primer paso es diseñar la pantalla principal, que es la página a la que accederán los usuarios cuando entren a la Web. Como se ha mencionado antes, se quiere que sea simple y que muestre los elementos básicos para poder jugar. Estos elementos son añadir un enlace a la página de configuración de partida, para que un usuario pueda crear una partida a su gusto para jugar contra otras personas o contra la máquina, y añadir un enlace a la página que permite unirse a una partida ya creada. Con esos dos elementos ya podríamos tener una página funcional en la que los usuarios pudieran jugar, pero como el tema que estamos tratando es un juego, se observó como suelen ser los menús de los videojuegos, y se optó por añadir otros elementos. Uno de ellos es muy útil, que es un enlace a una página en la que aparece el libro de instrucciones del juego, para que los usuarios que entren de nuevas a la página puedan aprender a jugar, o repasar las reglas para mejorar en el juego. Y el último elemento que se añade es un enlace a los créditos, tal y como hay en los juegos tradicionales, para reconocer a las personas involucradas en este proyecto. En la Fig. 19 se observa el diseño de esta pantalla.



Fig. 19. Esquema pantalla inicial página web

La implementación de la pantalla de créditos es más sencilla, ya que es una simple enumeración de las personas que han aparecido y organizaciones que han colaborado, cosa que no requeriría complejidad. Cuando el usuario clique en la opción de ¿Cómo jugar?, se le redirigirá a un PDF con el libro de instrucciones del juego de mesa, mismo PDF que aparece en este documento, en el anexo IV.

El siguiente enlace es el de unirse a una partida, que nos dirige a una página nueva. Es una pantalla bastante simple, ya que no se requieren de demasiados elementos para acceder a una partida, y de esta forma seguimos la simplicidad que queríamos lograr con esta página (Fig. 21). Se decide solo añadir los elementos necesarios que el usuario debe introducir para que acceda a una partida, para no sobrecargar la página y que el usuario no se pierda a la hora de intentar empezar a jugar una partida que ya esté creada.

Ya solo queda por explicar una pantalla a la que se puede acceder desde la pantalla principal, y esa es la de comenzar nueva partida. Esta pantalla va a ser más compleja que todas las anteriores, porque el usuario configurará una partida nueva con todas sus opciones desde ella. Es cierto que se podría haber separado esta configuración en distintos apartados que estuvieran en páginas distintas, pero se decidió incluirlo todo en una misma pantalla para otorgar al usuario de una vista global de cómo será la partida que se jugará.

En el centro de esta pantalla aparecerá el tablero con el que se jugará la partida. ¿Y por qué se incluye el tablero? Porque dependiendo de las opciones que escoja el usuario, el tablero tendrá distintos aspectos, y estos se actualizarán conforme el usuario realice cambios en las configuraciones. Así, esta vista previa del tablero de la partida va cambiando conforme se añaden nuevos jugadores o se cambian sus dimensiones.

A la izquierda, aparece una lista de jugadores. Se decidió reservar un espacio solo para la administración de los jugadores que iban a jugar en la partida, en primer lugar, porque así se diferencia esta parte de la configuración con la de los ajustes que tendrá la partida, y, en segundo lugar, porque tal y como está pensada, la altura de ese espacio va a ser variable, y es mejor que tenga un espacio propio para no desplazar a otras partes de la página. Se podría haber puesto en ese espacio una lista con 10 jugadores y que solo se rellenaran los datos de los que iban a jugar, pero de la forma en la que se ha hecho se ve de forma mucho más clara cuantos jugadores son los que jugarán la partida.

Lo último a explicar es el apartado de configuración de partida, que es el que aparece a la derecha. Se separa en este apartado todo lo relacionado con los ajustes de la partida, ya que la selección de estos elementos alterará el funcionamiento de la partida, tanto de las dimensiones del tablero como del modo de juego deseado, y al estar al final de la página (horizontalmente), el usuario tenderá a decidir estos ajustes una vez tenga claro quiénes van a jugar, algo que resulta más cómodo y que le dará una mayor perspectiva de cómo va a ser la partida.

Todo lo anteriormente explicado se ve reflejado en la Fig. 20, dónde se ve la pantalla de creación de nueva partida. Respecto al estilo, comentar que es similar al del resto de pantallas de la Web, respetando los colores identitarios de este juego. Mencionar que en esta pantalla se ven reflejadas la mayoría de mejoras que se han introducido en este juego respecto al juego original, ya que se proporcionan multitud de opciones de configuración que no existen en el juego de mesa debido a las limitaciones del aspecto físico. El tablero físico tiene un tamaño fijo que limita las dimensiones de la partida, y también limita el número de usuarios que pueden jugar la partida, aspectos que logramos superar con esta pantalla de configuración. Observando esta pantalla, nos cercioramos de logros que este trabajo aporta al juego, ya que no se limita simplemente a replicar el comportamiento del juego original

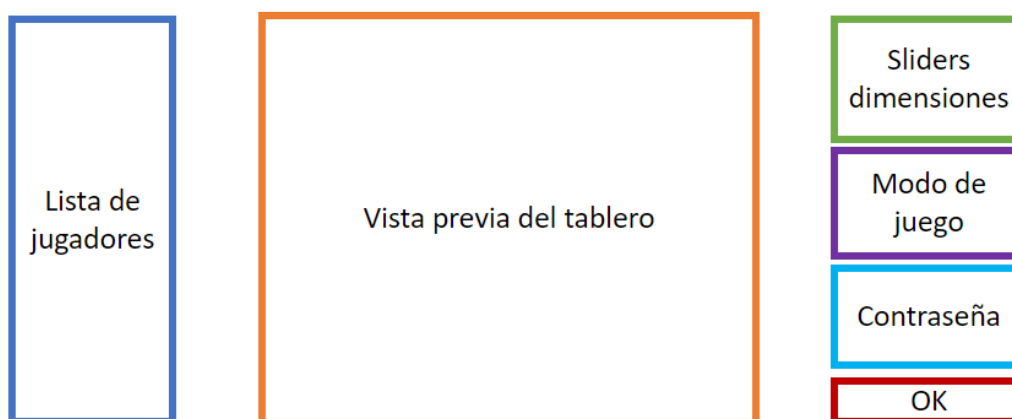


Fig. 20. Esquema pantalla de creación de partida

5.2.- Diseño pantalla de juego

La última pantalla que queda por explicar de esta Web es la que aparecerá después de esta, y es la de la propia partida. Ya se han observado pantallazos de ella en otros apartados, y se explican sus funcionalidades con más detalle en el manual de usuario (anexo I).

Destacar que la gran ventaja de usar una interfaz Web, es que se puede aportar mucha más información al usuario mientras juega la partida, información que no es tan sencilla de mostrar cuando se juega al juego de mesa físico. Por ejemplo, aquí las estadísticas se actualizan de forma automática, haciendo que el usuario no tenga que contar a mano lo que ha logrado en el transcurso de la partida, proceso que puede resultar muy tedioso cuando se juega al juego de mesa. También se muestran ayudas visuales para que el jugador sepa que bacterias puede expandir y a dónde expandirlas, o con ciertas cartas se muestra información gráfica para facilitar su uso, ya que cuando el tablero está muy poblado en el juego de mesa puede resultar muy confuso.

Existían ciertos problemas que solventar al diseñar esta pantalla. Por ejemplo, el tablero no tiene un tamaño fijo, porque depende de lo que se escoja en configuración. Eso se sortea aumentando el tamaño lo necesario. Quizás se pueda pensar que esto haga la página demasiado grande, pero se prefirió esto porque si se hubiera escogido una altura y anchura fijas, para tableros grandes las casillas podrían resultar demasiado pequeñas. Luego, hay situaciones de la partida que los jugadores comunicarían oralmente mientras juegan, como son el paso de turnos o marcar cuando han terminado ciertas acciones como la de expandir bacterias. Se

decidió reservar un espacio debajo del tablero para mostrar mensajes, así se indicaría al usuario qué es lo que debe hacer en cada momento y se le deja escoger cuando quiere terminar esa acción y pasar a la siguiente.

Con todo lo nombrado tenemos una idea de que necesitamos poner en la pantalla, pero ahora hay que escoger su distribución. Lo más importante de la partida, al fin y al cabo, es el tablero porque es dónde se desarrolla la partida. Por ello se coloca lo más arriba posible de la pantalla. Una vez colocado el tablero, el resto de elementos se distribuirán a su alrededor. La idea al hacer esto es que en la misma línea visual se sitúe el tablero y los elementos que ayudan al usuario a seguir el desarrollo de la partida. Por ello se coloca a un lado la lista de jugadores, con un indicador del turno actual, y al otro lado el objetivo que debe conseguir seguido de las estadísticas de la partida, para que el usuario compruebe si se acerca a ese objetivo o no.

Debajo de esa línea, aparecen los mensajes de información necesarios para desarrollar la partida. Se coloca ahí, centrado, para que sea muy visible para el usuario y que no se pierda en la partida.

Para finalizar, abajo del todo aparecen las cartas que el usuario tiene en la mano. Se reserva un espacio para poder colocar el máximo de cartas que tiene un usuario. El lugar escogido es ese porque se asemeja a como jugamos a juegos en la vida real, con el tablero delante de nosotros en la mesa y con las cartas cercanas a nosotros en nuestra mano. En nuestra pantalla la parte de arriba es la más lejana al jugador en la vida real, la mesa del juego, y la más cercana a nosotros es la de abajo, nuestra mano. En la Fig. 21 se observa un esquema con la distribución explicada de esta página. En el manual de usuario aparecerán pantallazos de la página Web para explicar cómo se usan todos los apartados.

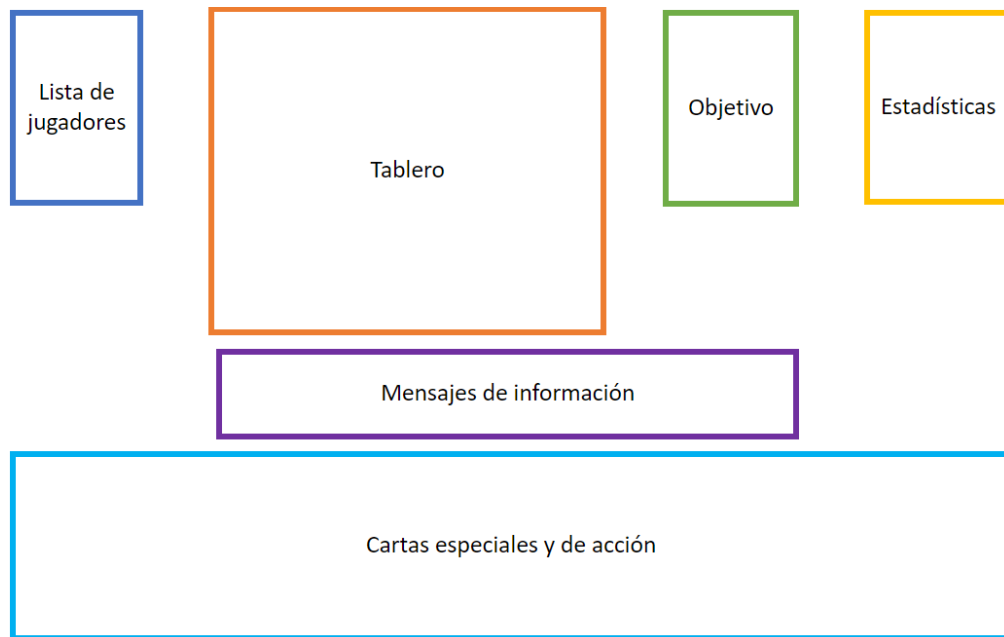


Fig. 21. Esquema pantalla de juego

6.- CONCLUSIONES

Se recibió un juego de mesa que estaba aún en proceso de ser terminado, ya que faltaban muchos elementos gráficos por crear y reglas por pulir, y se intentó adaptarlo a un entorno web. Hubo que estudiar y analizar el juego de mesa, comprender sus reglas y funcionamientos para poder crear la página web basada en él. Se programó con HTML, CSS y JavaScript todas sus funcionalidades, y con diversas librerías de JavaScript se añadió la funcionalidad de comunicación entre cliente y servidor. A mitad del desarrollo del trabajo, se recibieron actualizaciones del juego de mesa, y una vez finiquitado éste, se pudo empezar a adaptar de forma fiel el juego de mesa, cumpliendo los propósitos con los que se partían a la hora de empezar este TFG. De esta forma, con la página web ya creada se conseguirá aportar a la divulgación científica, que era el objetivo principal del juego de mesa, proporcionando una forma más sencilla de acceder a jugar a este juego, al no estar limitados por la necesidad de tener que adquirir el juego de mesa.

El desarrollo de este trabajo no ha sido coser y cantar, ya que consta de bastantes conocimientos nuevos que han tenido que ser aprendidos en el transcurso del trabajo. Las principales afrontadas han sido:

- Aprender HTML y CSS: Nunca se había programado en Web, así que no sabía usar estos lenguajes. Tuve que aprender desde cero a usarlos, y el aspecto del diseño de la página y como distribuir los objetos en ella quizás fue el más complicado de todos, porque no son lenguajes muy intuitivos de cara al usuario (sobre todo CSS). No incluyo a JavaScript en esta lista, porque al fin y al cabo es un lenguaje de programación, y es más fácil adaptarse a él gracias a las bases de programación adquiridas en la carrera.
- Encontrar protocolo de comunicación: Hallar el sistema a través del cual se iba a desarrollar el juego online costó más de lo esperado, porque eran muchas las opciones disponibles, y no se podía empezar a diseñar el juego online hasta tener esto claro, algo que pudo paralizar el progreso del trabajo. Al final, se logró hallar un sistema simple pero efectivo, por lo que se está satisfecho con el resultado final.

Además, los elementos gráficos con los que se partía en este trabajo eran bastantes pocos, porque aún no estaba finalizado el juego de mesa, pero a mitad del transcurso del trabajo llegaron los diseños definitivos. Aun así, luego se decidieron añadir las mejoras explicadas en

el capítulo 4, y no se disponía de elementos gráficos suficientes para satisfacer las necesidades del programa porque el juego de mesa estaba pensado para cuatro jugadores como máximo, y en este trabajo se implementan hasta 10 jugadores, por lo que el material gráfico usado por esos nuevos jugadores es material reciclado de otros jugadores.

6.1.- Desarrollo del trabajo e incidencias

En cuanto al desarrollo del trabajo, cabe destacar que ha llevado más horas de las que se pretendía en un principio (300 horas). Esto es debido a la falta de conocimiento previo en los aspectos que trataba el juego, sobre todo en los elementos de programación usados, y además porque se le ha querido dar un aspecto más profesional, intentando buscar un diseño más completo y una cercanía estilística al juego de mesa. Y en principio estaba previsto entregarlo en septiembre, pero la fase de la creación de juego online llevó más tiempo de desarrollo del esperado, y no por la propia programación en sí, sino la fase de análisis y elección de herramientas y protocolos que usar, ya que se tuvo que desechar parte del trabajo hecho para volver a empezar de nuevo con un sistema mejor. A continuación, se mostrará un diagrama de Gantt con el desarrollo del trabajo (Fig. 22), y después se añadirá un desglose de las horas que cada fase ha requerido.

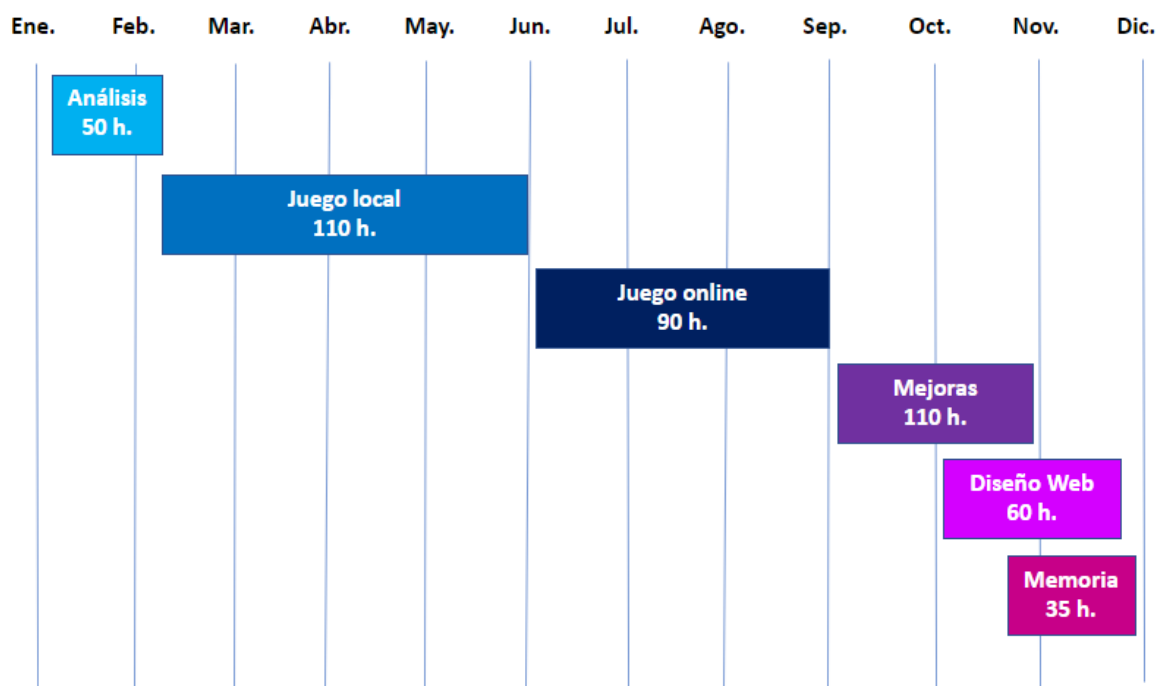


Fig. 22. Diagrama de Gantt con el desarrollo del trabajo, desde enero de 2021 a noviembre de 2021

- Fase 1, análisis: horas empleadas: 50. Esta fase consiste en el análisis del trabajo que se debía hacer, estudiar el juego de mesa, sus reglas y elementos de la partida, y aprender cómo funcionan los lenguajes informáticos que usar. Se detalla a continuación el trabajo realizado en esas horas:
 - Estudiar HTML, CSS y JavaScript: 35 horas. En esta parte se estudiaron los elementos básicos de cada lenguaje. No se pretendía aprender todo sobre estos lenguajes, ya que muchos de esos conocimientos se irían adquiriendo con la práctica y conforme se realizara el trabajo, pero si se estudiaron diversos ejemplos de juegos hechos con esos programas para tener una base con la que empezar a desarrollar.
 - Estudiar juego de mesa: 15 horas. Aprenderse el manual de instrucciones que había sido suministrado en ese momento (todavía no era la versión final). Cuestionarse como funcionaba el juego y preguntar dudas a las creadoras para afianzar el conocimiento y comenzar a desarrollar el juego de forma correcta.
- Fase 2, juego local: horas empleadas: 110. Esta fase consiste en el desarrollo del juego a nivel local, para asegurarse de que puede funcionar y para mostrar cómo sería el desarrollo de la partida, además de para adquirir cierta soltura en la programación con JavaScript y HTML. Se detalla a continuación el trabajo realizado en esas horas:
 - Estudio de tableros hexagonales: 30 horas. Fase que es más complicada de lo que parece, porque el diseño del tablero era muy concreto y de él dependía todo el funcionamiento del juego. Hubo que buscar como crear y dibujar estos tableros en JavaScript, diseñar el sistema de coordenadas que regiría el tablero y crear funciones para manejar la interacción entre el usuario y el tablero [3]. Una vez encontrada la fuente que explicaba estos procesos, el desarrollo de esta fase del trabajo se simplificó bastante.
 - Análisis y diseño de la partida: 20 horas. En este momento principalmente se pensó como iba a desarrollarse la partida, como iba a diseñarse el juego y los elementos básicos que componen una partida. Se decidió el tipo de interacciones que se quería que el usuario tuviera, y como se iban a representar las acciones del usuario en el juego.

- Implementación: 60 horas. Último apartado de esta fase, consistente en la programación de todo el juego en JavaScript siguiendo lo diseñado anteriormente. Al final de este proceso ya se podían jugar partidas al juego desde un mismo equipo.
- Fase 3, juego online: horas empleadas: 90. En esta fase se halla el protocolo a usar para la comunicación entre cliente y servidor, e implementar ese sistema de mensajes en el juego local anteriormente hecho, para poder jugar ya partidas entre jugadores remotos.
 - Hallar protocolo de comunicación: 50 horas. Quizás pueda parecer excesivo, pero costó decidirse por el protocolo ideal, que terminó siendo WebSocket a través de la librería socket.io. Al principio se estudiaron otros protocolos, e incluso se tuvo que desechar trabajo hecho al no haber escogido correctamente el sistema a usar. Luego se diseñaron los tipos de mensajes que iba a usar nuestro programa.
 - Implementación: 40 horas. Escribir en código el sistema de mensajes escogido, y después adaptar la implementación hecha a nivel local añadiendo esos mensajes dónde fuera oportuno. Además, hubo que cambiar cómo funcionaba el juego, ya que ahora cada jugador jugaría en una pantalla distinta, cuando en el prototipo a nivel local lo hacían todos desde la misma pantalla.
- Fase 4, mejoras al juego original: horas empleadas: 110. Fase en la que se idean todas las mejoras que se quieren añadir al juego (algunas ya se sabían, como las de las IAs o partidas a tiempo real). Después se estudia cómo se van a implementar y finalmente se añaden a todo el código del juego online hecho previamente.
 - Establecer mejoras que añadir: 10 horas. Pensar y poner por escrito que cosas se quieren implementar al juego. Algunas de ellas ya estaban pensadas con anterioridad, así que aquí van mejoras como la de aumentar el número de jugadores, creación dinámica de tableros o variar las dimensiones del tablero.
 - Diseño de jugadores controlados por la máquina: 35 horas. En esta fase se piensa el proceso de actuación que van a seguir, decidir las probabilidades que tendrán asignadas cada una y después su implementación en el juego, para que sean capaces de realizar las mismas acciones que realizaría un humano.
 - Partidas en tiempo real: 15 horas. Aquí lo que se hizo es pensar cómo se iba a implementar en el juego y ejecutar esas ideas. Quizás parezcan pocas horas, pero el protocolo de comunicación usado facilitó mucho la implementación de este modo

de juego, ya que al final es una adaptación del modo de juego normal, el que es por turnos.

- Creación de nuevos tableros y ampliación de número de jugadores: 50 horas. El juego en principio estaba pensado para ser jugado de dos a cuatro personas, como en el juego de mesa original, pero se decidió que podíamos aprovechar la plataforma realizada para ampliar esas posibilidades, e incluir partidas con un mayor número de jugadores. Hubo que rehacer parte del código, porque éste estaba diseñado para cuatro personas, y además hubo que generar el código y algoritmos que se encargan de la creación de nuevos tableros, porque el mayor número de jugadores requería de nuevos tableros. Se diseñó la asignación dinámica de casillas de salida y casillas bacteriostáticas, y se decidió prescindir de tableros preestablecidos porque limitaban bastante el crecimiento del programa.
- Fase 5, diseño de la página: 60 horas. Finiquitar el diseño de la página web. Aunque muchas de los elementos que había que poner ya estaban decididos por los apartados anteriores, aquí se centró el trabajo en darle un aspecto aceptable a la página Web, y a organizar bien las páginas que formaban la página web.
 - Pensar diseño Web: 15 horas. Establecer el diseño que iban a tener los distintos apartados de la página Web.
 - Implementación del diseño: 15 horas. Empezar a implementar las ideas pensadas con anterioridad, acercando el aspecto de la página web a lo deseado.
 - Revisiones y correcciones al diseño: 30 horas. Después de reuniones con el director de trabajo, se corrigen errores de la página Web y se pulen los diseños, gracias al feedback que aportó. Hay múltiples correcciones al diseño y varias revisiones al diseño.
- Fase 6, elaboración de la memoria: 35 horas. Documentación y posterior elaboración de la memoria del trabajo y anexos correspondientes.

6.2.- Posibles ampliaciones al trabajo

Aunque el trabajo este completado, es deseable también estudiar las formas en las que se podría ampliar este trabajo. Quizás la más evidente reside en el apartado gráfico de la página, ya que estaba limitado por el material gráfico que fue proporcionado por las creadoras del juego de mesa, y para aspectos como el de añadir más de cuatro jugadores hubo que reutilizar algunos de esos materiales. Así que el resumen sería que el siguiente paso en este trabajo sería encargar más material gráfico para cubrir las necesidades creadas por el juego, y también para mejorar el aspecto de la página, ya que este trabajo se centró en crear la estructura web suficiente para jugar al juego, pero no incluyó el estudio y creación de imágenes o banners por falta de tiempo y de conocimiento sobre diseño gráfico.

Otra posible mejora sería añadir gestión de cuentas de usuario, para que se pudiera almacenar allí las diversas estadísticas que ha tenido a lo largo de sus partidas, y facilitar el proceso de invitación a partidas. Esto también permitiría añadir clasificaciones en la página web que podrían estimular que se jugara más al juego.

Cabe destacar que se podrían añadir nuevos comportamientos de los jugadores controlados por la máquina cambiando las probabilidades que hemos establecido, algo que alteraría su forma de jugar. También decir que se podrían haber añadido mejoras a los algoritmos si se hubiera dispuesto de más tiempo en el trabajo. Una que quizás sea evidente es implementar una función que hiciera al algoritmo centrarse en un jugador rival al que destruir sus bacterias, centrando sus esfuerzos en él. Esto sería útil si el programa detecta que un jugador tiene demasiadas bacterias en el tablero, y quisiera frenar su avance por si acaso el rival tuviera un objetivo de conquistar casillas. Otra mejora posible sería hacer que las bacterias del jugador controlado por la máquina se expandieran en una dirección concreta, por si su objetivo es llegar a una casilla de salida, o por si posee bacterias con genes de resistencia y quiere frenar el avance de un jugador rival a cierta parte del tablero. Y como estas mejoras que se proponen, con tiempo para pensar el diseño de estos comportamientos y realizar su implementación se podrían conseguir jugadores con mucha mayor complejidad, pero como este trabajo incluía muchos más objetivos, se decidió realizar algoritmos funcionales y básicos que sirvieran para el propósito de permitir partidas sin jugadores humanos rivales.

6.3.- Experiencia adquirida y opinión personal

Aunque las horas invertidas en este proyecto puedan haber sido superiores a las deseadas en un principio, es cierto que ha proporcionado una buena base de conocimiento en programación web, ya que se ha adquirido conocimiento esencial en HTML y JavaScript que no se tenía al empezar este trabajo. Esto es algo que teniendo en cuenta la rama del grado en la que nos encontramos, Telemática, puede ser bastante útil de cara al futuro, porque permite conocer otra parte de la comunicación en Internet, más cercana a las capas superiores del modelo OSI que no hemos tratado tanto en este grado.

A pesar de lo anterior, también hay que destacar que se han aplicado muchos conocimientos adquiridos en la carrera, ya que estos han sido vitales para poder conseguir la comunicación entre los distintos clientes que accedan al juego. También para poder programar el juego de forma correcta, porque todos los conocimientos de programación aprendidos durante el grado han sido aplicados. Es cierto que los lenguajes de programación no habían sido usados en el grado, pero gracias a la base de programación adquirida durante estos años, se ha sido capaz de usar nuevos lenguajes sin que la adaptación fuera dramática.

Para finalizar, destacar que se está bastante satisfecho con todo el trabajo hecho y lo que se ha logrado, y también con las personas envueltas en el desarrollo, como son el director de este trabajo, con el que la comunicación siempre ha sido fluida y todos los aportes que ha hecho han sido bastante interesantes. También mencionar a las creadoras del juego original de mesa, que han estado siempre muy receptivas a los progresos realizados y han mostrado una actitud muy positiva para con el proyecto.

BIBLIOGRAFÍA

[1] El proyecto de divulgación Bacterfield comenzó su desarrollo en enero de 2020. Recibió financiación de la Fundación General CSIC (FGCSIC), a través de la convocatoria Cuenta la Ciencia (3ª edición). Además, algunos elementos del proyecto han sido financiados por un proyecto Marie Skłodowska-Curie de la comisión europea (MSCA-IF 2018 845427). <https://www.heraldo.es/noticias/aragon/2021/05/03/investigadoras-del-inma-crean-bacterfield-un-juego-de-mesa-que-permite-dar-a-conocer-el-mundo-de-la-microbiologia-1489321.html> (accedida 19 de noviembre, 2021)

[2] “Usage statistics of JavaScript as client-side programming language on websites” <https://w3techs.com>. <https://w3techs.com/technologies/details/cp-javascript/> (accedida 30 de octubre, 2021)

[3] A. Patel. “Hexagonal Grids from Red Blob Games” [redblobgames.com](https://www.redblobgames.com). <https://www.redblobgames.com/grids/hexagons/> (accedida 10 de marzo, 2021)

[4] Carlos Azauste. “WebSockets: Cómo utilizar Socket.io en tus aplicaciones web” carlosazauste.es <https://carlosazauste.es/websockets-como-utilizar-socket-io-en-tu-aplicacion-web> (accedida 25 de agosto, 2021)

[5] Bacterfield participa en la feria Science is Wonderful y es accésit en los premios Tercer Milenio. http://www.unizar.es/actualidad/vernoticia_ng.php?id=63065&idh=11370&pk_campaign=iunizar20211123&pk_source=iunizar-personalunizar (accedida 23 de noviembre, 2021)

ANEXO I – MANUAL DE USUARIO

En este manual de usuario se va a describir como navegar a través de la página de Bacterfield y cómo utilizar todas sus funcionalidades.

I.1.- Página principal



Fig. 23. Pantalla principal

Como vemos en la anterior imagen (Fig. 23), la portada tiene 4 apartados distintos:

- Comenzar nueva partida: Permite al usuario crear una partida propia a su gusto.
- Unirse a una partida: Permite al usuario unirse a una partida ya existente.
- ¿Cómo se juega?: Se explica el funcionamiento de la página y las reglas del juego.
- Créditos: Listado de personas envueltas en la creación del juego.

I.2.- Unirse a una partida



Fig. 24. Pantalla para unirse a una partida

Aquí (Fig. 24) el usuario puede unirse a una partida ya existente. Puede escoger el nombre que tendrá en la partida, y deberá introducir la contraseña de la partida a la que quiere acceder. Además, en el desplegable de tipo de jugador podrá escoger si juega él (opción de Humano) o si una inteligencia artificial lo hará en su lugar (distintas opciones de IAs). Si la contraseña introducida es incorrecta, aparecerá un mensaje de error indicándolo. Lo mismo sucederá si la partida a la que se intenta acceder no tiene huecos restantes.

I.3.- Comenzar nueva partida



Fig. 25. Pantalla para unirse a una partida

Pantalla en la que el usuario puede crear una partida (Fig. 25). Dispone de 3 elementos diferenciados como se puede observar. El primero de ellos es la lista de jugadores que participarán en la partida, el segundo de ellos una vista previa del tablero en el que se jugará la partida, y el último de ellos es los aspectos de configuración que tendrá nuestra partida.

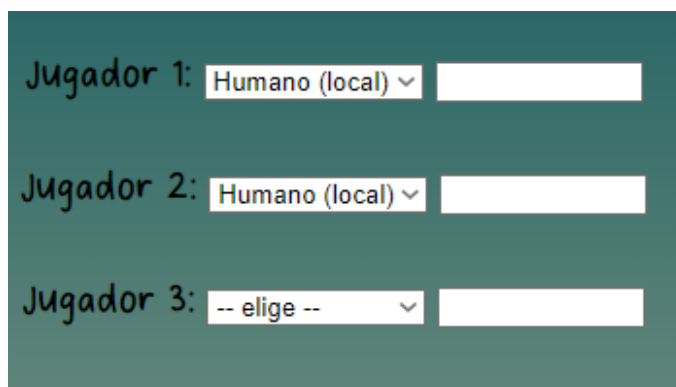
The image shows a user interface for selecting players. It has a dark green background. There are three rows, each for a player. Each row consists of a label (Jugador 1, 2, 3), a dropdown menu, and a text input field. The first two rows have 'Humano (local)' selected in the dropdown. The third row has '-- elige --' selected. The text input fields are empty.

Fig. 26. Apartado de selección de jugadores

Aquí se puede escoger el número de jugadores (Fig. 26). Se hace abriendo el desplegable del último jugador de la lista, y escogiendo el tipo de jugador que queramos. Las distintas opciones son:

- Humano (local): jugador que jugará de forma local, la página abrirá una pestaña nueva en el navegador en la que podrá jugar.
- IA1, IA2, IA3: también juegan a nivel local, como la anterior opción, pero lo harán de forma automática controladas por las IAs establecidas en el juego. La IA1 actúa completamente al azar, la IA2 lo hace siguiendo el objetivo ciegamente, y la IA3 es similar a la anterior, pero con mayor margen de maniobra, no condenada a seguir el objetivo.
- Online: se utiliza esta opción si se quiere crear un hueco en la sala para que acceda un jugador de forma remota, y él lo hará usando el apartado de unirse a una partida que aparece en la pantalla principal.

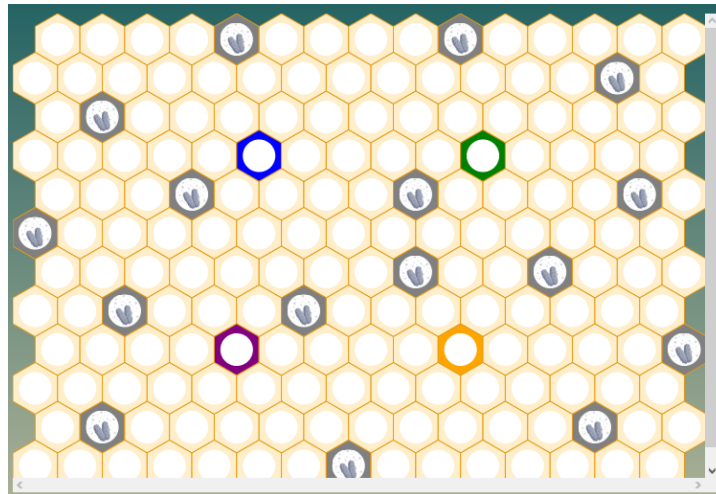


Fig. 27. Tablero creado

Como ya se ha dicho antes, en este apartado tenemos un vistazo a cómo será el tablero de la partida (Fig. 27). Pero tiene la peculiaridad de que conforme se añadan nuevos jugadores, el tablero se irá adaptando para darles cabida. Si deseamos jugar con menos jugadores, deberemos refrescar la página e iniciar de nuevo el proceso añadir jugadores.

Dimensiones

Altura: 12

Anchura: 15

Tipo de partida

☒ Por turnos

☐ Tiempo real

Contraseña:

OK

Fig. 28. Apartado de selección de elementos de la partida

Este es el último apartado de esta página (Fig. 28), y permite seleccionar las características que queremos que tenga nuestra partida.

En dimensiones, podemos escoger el ancho y altura del tablero de la partida, desplazando el slider hacia derecha o izquierda.

En tipo de partida, escogemos si queremos que la partida sea por turnos, más fiel al juego de mesa, o funcione a tiempo real, con todos los jugadores realizando acciones a la vez.

Por último, tenemos el apartado de contraseña, donde introduciremos una clave asociada a nuestra partida. Con ella podrán unirse nuevos jugadores a nuestra partida de forma remota, o si algún jugador se desconecta en medio de la partida, puede ir al apartado de unirse a una partida e introducir la contraseña para regresar a ella. Es posible iniciar una partida sin introducir una contraseña, pero hay que tener en cuenta que esto impedirá a los jugadores incorporarse a ella, ya que no habrá clave que usar para ello.

Cuando se le dé al botón de OK, la partida se creará. El usuario debe permitir al navegador mostrar ventanas o pestañas emergentes para que funcione el juego.

I.4.- Pantalla de partida



Fig. 29. Pantalla de partida una vez creada

Cuando creamos la partida, accederemos a esta pantalla (Fig. 29). Si somos un jugador que se ha unido a la partida, lo único que podremos hacer es esperar a que empiece. Si somos el jugador que la ha creado, podremos darle comienzo clicando en el botón azul de START.

Cabe notar que a la izquierda aparece la lista de jugadores, marcando que jugador somos en rojo. Aunque hayamos creado una partida con x jugadores, si le damos a empezar partida sin que se hayan unido todos, ésta empezará, así que conviene esperar a que todos se unan si no se quiere dejar a nadie fuera. Conforme los jugadores se unan, sus nombres aparecerán en la lista de jugadores.

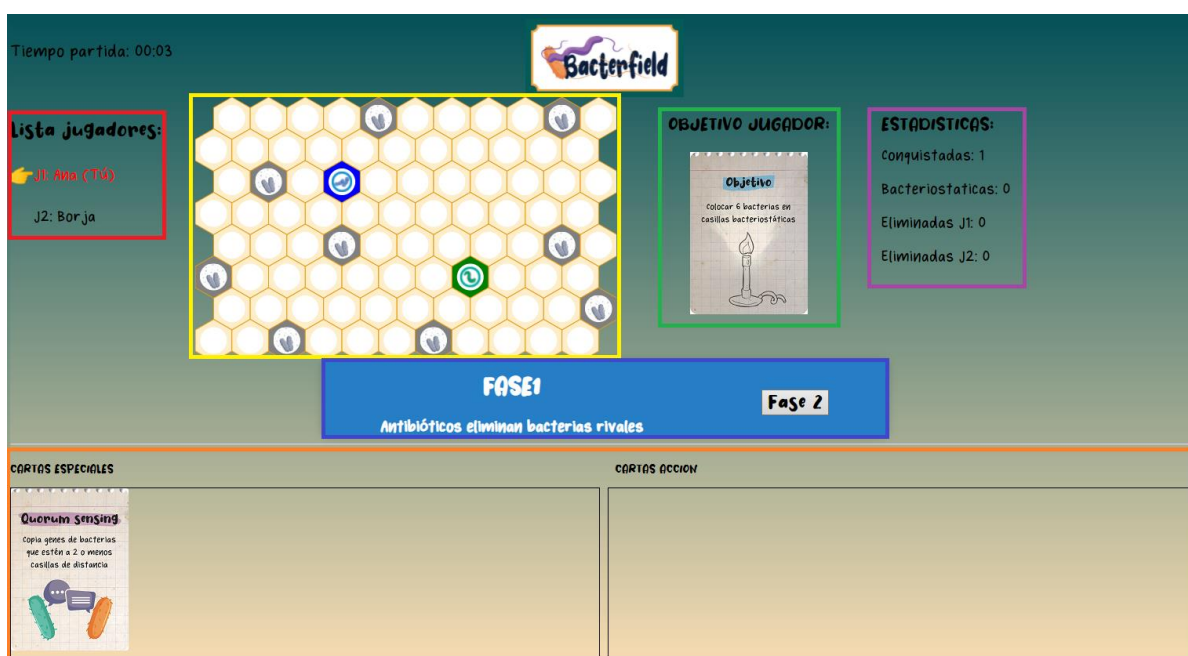


Fig.

30. Pantalla de partida una vez iniciada, señalizada

En la Fig. 30 se observa cómo es la pantalla de juego cuando ha empezado la partida. Se han señalado con recuadros de colores los distintos apartados que la componen, y que se van a enumerar a continuación.

En el recuadro rojo está la lista con los jugadores que se encuentran en partida. Se marca de quien es el turno con el icono de la mano apuntando, y el jugador que está jugando en esta pantalla en rojo. Si un jugador se desconecta, desaparecerá su nombre y aparecerá “Desconectado” en su lugar, en color gris.

En el recuadro amarillo está el tablero de la partida, los jugadores tendrán que interactuar con él para mover o eliminar fichas, o cambiar los genes de las bacterias deseadas. Todo el desarrollo de la partida sucede aquí, y es un tablero que comparten todos los jugadores y que se va actualizando de forma simultánea.

En el recuadro verde aparece el objetivo que tendrá que conseguir el jugador, y se le asigna cuando se une a la partida repartiéndole una carta del mazo. Existen diversos objetivos que aparecen explicados en las instrucciones del juego, del anexo IV.

En el recuadro morado aparecen las estadísticas que lleva el jugador en la partida, hay que tenerlas en cuenta porque cuando se alcanzan diversos hitos, se puede cumplir el objetivo asignado y resultar ganador de la partida. Las estadísticas que aparecen son:

- Conquistadas: Celdas ocupadas por bacterias propias. Si la bacteria está infectada con virus, no contará para esta estadística. Estadística vital para el objetivo de conquistar 48 casillas.
- Bacteriostáticas: Celdas bacteriostáticas (celdas grises) ocupadas por bacterias propias (que no estén infectadas por virus). Estadística vital para el objetivo de conquistar 6 casillas bacteriostáticas.
- Eliminadas J1-J2-...-JN: Bacterias eliminadas de cada jugador (J1 de jugador 1, J2 de jugador 2, etc.). Estadística que hay que tener en cuenta para el objetivo de eliminar 24 bacterias rivales, o eliminar 8 bacterias de cada jugador.

En el recuadro azul aparecerán los mensajes que permitirán al usuario ejecutar las distintas fases que componen un turno, avanzando a través de ella y eligiendo que acciones hacer.

En el recuadro naranja aparecen las cartas que tiene el jugador en la mano. Puede tener 5 como máximo. Si quiere usarlas, tendrá que clicar sobre ellas, y luego efectuar la acción que desee.

A continuación, se explicarán algunos apartados un poco mejor, y como deben interactuar los usuarios con ellos para avanzar en la partida.



Fig. 31. Mensaje de fase 2 del turno

Fase 2 del turno (Fig. 31), aquí el usuario puede usar una carta especial que tenga en la mano. Para usarla, clicas en la carta, y luego en la bacteria en la que quiera aplicar el efecto. Después de usar una carta especial, debe escoger que hará en la siguiente fase, bien utilizar una carta de acción o bien expandir sus bacterias. Si escoge carta de acción, podrá usar una carta de acción que tenga. Si no, pasará a expandir bacterias.

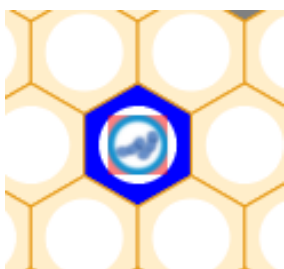


Fig. 32. Fase de expansión de bacterias



Fig. 33. Elección de expansión de bacteria

Aparecen con el fondo rojo las bacterias que podemos expandir en nuestro turno. Clicamos en ella si queremos expandirla. Observamos el aspecto que tiene la ficha en la Fig. 32.

Cuando clicamos la bacteria que queremos expandir, se marcan con círculos rojos las casillas a las que podemos expandirnos (Fig. 33). Clicamos la casilla que queramos ocupar, y la nueva bacteria se colocará allí.

Con esto se ha explicado todo lo que se puede hacer en un turno. La partida procederá, y cuando un jugador logre su objetivo, se terminará, apareciendo un mensaje en pantalla indicando el jugador ganador de la partida.

ANEXO II – PUESTA EN MARCHA DE SERVIDOR Y PÁGINA WEB

Como se ha explicado en la memoria, para usar el servidor es necesario disponer de las librerías Node.js, Express.js y socket.io. En la siguiente figura (Fig. 34) se muestra la estructura a través de la cual funciona esta página web, con un equipo que hace de servidor con todos los archivos necesarios, y una serie de equipos remotos que acceden a él a través de su IP y el puerto en el que esté escuchando las conexiones.

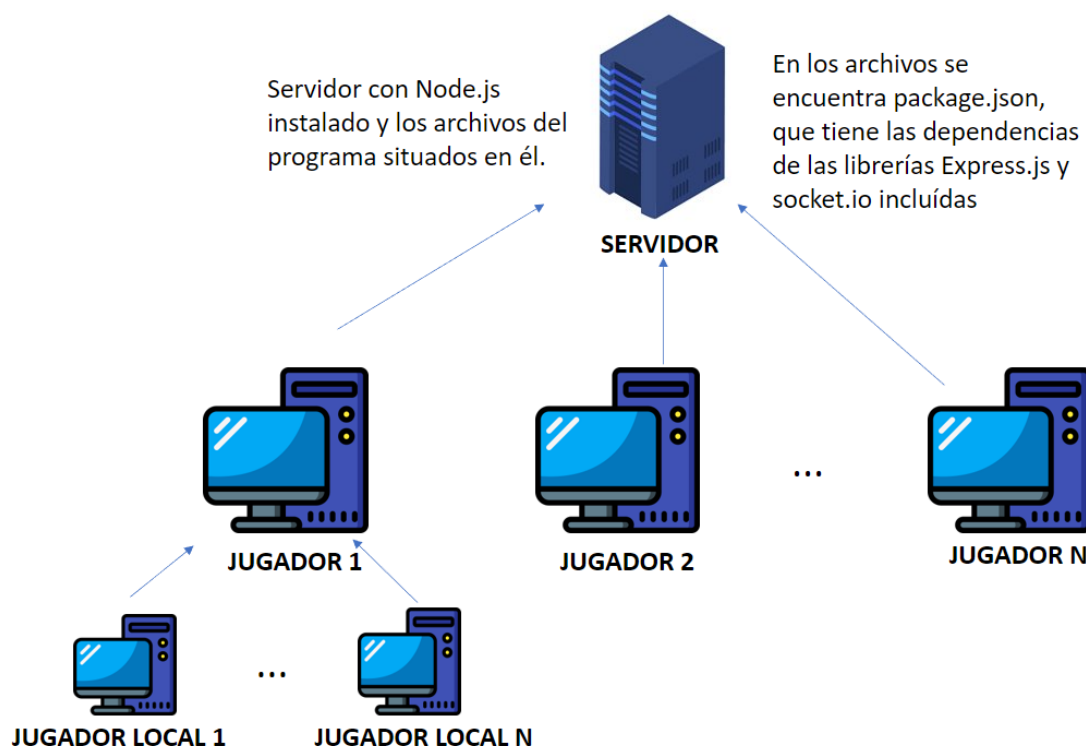


Fig. 34. Estructura que muestra cómo está disponible el juego Web

Ahora se explicará cómo se puede poner en marcha esta Web en el servidor que queramos. Lo primero será escoger la dirección en la que lo queremos albergar, y luego colocar allí los archivos que permiten funcionar a la Web. Estos archivos se proporcionarán en un link que redirigirá a Google Drive, estando el link situado al final de este anexo.

Lo segundo será descargar Node.js en el equipo que hace de servidor para esta página Web. Cuando instalemos Node.js, nos vendrá incluido npm, el sistema de gestión de paquetes de Node.js. Con él podremos instalar las dependencias de las librerías Express.js y socket.io. Con

la consola de comandos accedemos a la carpeta en la que tengamos todos los archivos de nuestro programa, y ejecutamos los siguientes comandos:

`npm install socket.io --save`

`npm install express --save`

De esta forma ya tenemos nuestras librerías listas para ser usadas por el servidor, habiéndose guardado en las dependencias de nuestro archivo package.json.

Lo siguiente es indicarle a nuestro programa de servidor en que puerto debe estar escuchando para las conexiones entrantes, y eso lo podemos hacer entrando al archivo puerto.txt que encontramos en la carpeta server, lugar donde está el programa que funciona como servidor. En ese fichero de texto simplemente escribimos el número de puerto deseado y ya está (Fig. 35).

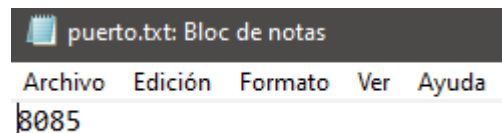


Fig. 35. Ejemplo de fichero puerto.txt, en este caso el servidor escuchará en el puerto 8085

Por último, basta ejecutar el programa, y eso también lo haremos por línea de comandos. Nos colocamos en la carpeta superior de nuestro programa, y ejecutamos la siguiente línea:

`node server/server.js`

Y con esto ya tendremos a nuestro servidor y página web corriendo y funcionando.

Enlace con los archivos necesarios para poner en marcha el servidor:

<https://drive.google.com/drive/folders/1TMA5I72Omm7XascXG2VsovFGNI2Vfjcs?usp=sharing>

ANEXO III – SISTEMA DE MENSAJES SERVIDOR-CLIENTE

En este anexo se enumerarán todos los mensajes que componen el sistema de este juego, y que permiten comunicarse a servidor y jugadores. Existe un tipo de mensajes que son los que el servidor emite hacia los clientes, y luego los mensajes que los clientes emiten hacia el servidor.

III.1.- Mensajes de control

Estos mensajes, intercambiados entre servidor y jugadores, se encargan de crear partidas, aceptar nuevos jugadores en ellas, intercambiar datos de la partida, gestionar desconexiones y terminar la partida.

- crearpartida: Mensaje enviado por el jugador que crea la partida. Envía en él los datos necesarios para crear la partida y establecerla. Estos datos son:
 - Contraseña: Contraseña de la sala, con ella el servidor crea una sala a la que asignar todos los jugadores que llegan, y para crear un índice en el que guardar todos los datos de la partida, que son los que se enumeran a continuación.
 - Tipo de partida: Si la partida es por turnos o en tiempo real.
 - Número de jugadores: Cantidad de jugadores que hay en la partida.

El servidor guarda estos datos, en los espacios que ha reservado para los datos de esta sala de juego (relacionada con la contraseña). Si la contraseña se envía vacía, se asigna como clave un número entero aleatorio que sirve de identificador de la sala, sustituyendo a la contraseña como tal. Después, crea una baraja de cartas para esta partida, y mezcla las cartas, listo para cuando tenga que mandar cartas al usuario.

- pasarceldas: Mensaje enviado por el jugador que crea la partida. Sirve para enviar al servidor el tablero con el que se va a jugar la partida. El servidor guardará este tablero y lo asignará a la sala con el identificador de la partida.

- nombrescomienzopartida: Mensaje enviado por el jugador que crea la partida. Aquí pasa los nombres de los jugadores que ha elegido el usuario que ha creado la partida. El servidor los guarda para enviárselos a los jugadores que se unan de forma remota.
- encontrarpartida: Mensaje enviado por un jugador que se quiera unir a la partida. Manda el mensaje con una contraseña, para ver si puede unirse a esa partida.
- partidaencontrada: Mensaje de respuesta del servidor al cliente que se quiere unir a la partida, en caso de que pueda unirse a ella.
- errorcontrasena: Mensaje de respuesta del servidor al cliente que se quiere unir a la partida, en caso de que no exista ninguna partida con la contraseña que ha introducido.
- errorsalallena: Mensaje de respuesta del servidor al cliente que se quiere unir a la partida, en caso de que no haya hueco para nuevos jugadores en la sala a la que desea acceder el jugador.
- unirsepartida: Mensaje enviado por el jugador que quiere unirse a la partida, después de recibir el mensaje de “partidaencontrada”. En el mensaje va incluido la contraseña de la partida a la que quiere acceder.
- asignar: Mensaje de respuesta del servidor al cliente que se quiere unir a la partida, en él el servidor le indica al cliente que jugador es (número de jugador), le manda las cartas con las que empezará la partida (la carta de su objetivo), le manda que tipo de partida es, cuantos jugadores hay en la partida y una variable con las bacterias que ha destruido cada jugador (al principio será vacía, pero si un jugador se une a mitad de partida, no estará vacía).
- datosceldas: Mensaje de respuesta del servidor al cliente que se quiere unir a la partida, en él le pasa el tablero con el que se va a jugar la partida.
- mandarnombre: Mensaje del jugador que se une a la partida, en este caso jugador remoto, para que el servidor sepa su nombre y lo almacene.
- message-actualizarnombres: Después de recibir el mensaje anterior, el servidor envía a todos los jugadores de la sala este mensaje, para que actualicen su lista de jugadores con el nuevo nombre.
- comenzarpartida: Mensaje enviado por el jugador que ha creado la partida, que indica al servidor que dé comienzo la nueva partida. El mensaje se envía cuando el jugador pulsa el botón de start.

- empezarpartida: Mensaje del servidor como respuesta al anterior mensaje, lo envía a todos los jugadores de la sala para que la partida empiece y todos se pongan a jugar.
- disconnect: Mensaje que se genera cuando un jugador se desconecta de la partida. El servidor eliminará a este cliente de su lista de jugadores de la sala, aunque guardará sus datos por si se vuelve a reconectar.
- desconectado: Mensaje del servidor a todos los clientes de la sala, informando de que jugador se ha desconectado de la sala, para que salten su turno si llega a él.
- gameover: Mensaje enviado de un cliente al servidor, informando de que ha ganado la partida al alcanzar su objetivo.
- messages-gameover: Mensaje enviado del servidor a todos los clientes de la sala, informando de que la partida ha terminado y diciendo que jugador es el que ha ganado.

III.2.- Mensajes de gestión de partida

Una vez ya listados los mensajes de control, que permiten crear la partida y que los jugadores se unan o desconecten de ella, ahora falta por listar los mensajes para que funcione el transcurso de la partida, en los que los jugadores se comunicarán con el servidor para actualizar el tablero de todos los jugadores. Todos los mensajes de los clientes llevarán el identificador de partida incluido, para que el servidor sepa de que sala es el mensaje y pueda responder solo a los jugadores de esa sala.

- new-message: Mensaje enviado de un jugador al servidor, informando de que ha añadido una ficha de bacteria al tablero. Le envía la casilla en la que ha añadido la bacteria y el número de jugador que es.
- messages: Mensaje del servidor a todos los clientes de la partida, se envía como respuesta al anterior mensaje. Les envía la casilla en la que añadir la ficha de bacteria, y de qué jugador es la bacteria a añadir.
- biofilm: Mensaje enviado de un jugador al servidor, informando de que ha añadido fichas de biofilm alrededor de una bacteria. Le envía la casilla en la que ha añadido las fichas de biofilm y el número de jugador que es.

- messages-biofilm: Mensaje del servidor a todos los clientes de la partida, se envía como respuesta al anterior mensaje. Les envía la casilla en la que añadir las fichas de biofilm, y de qué jugador es las fichas de biofilm a añadir.
- antibiotico: Mensaje enviado de un jugador al servidor, informando de que ha añadido una ficha con genes de antibiotico al tablero. Le envía la casilla en la que ha añadido la ficha y el número de jugador que es.
- messages-antibiotico: Mensaje del servidor a todos los clientes de la partida, se envía como respuesta al anterior mensaje. Les envía la casilla en la que añadir la ficha de antibiotico, y de qué jugador es la ficha a añadir.
- resistencia: Mensaje enviado de un jugador al servidor, informando de que ha añadido una ficha con genes de resistencia al tablero. Le envía la casilla en la que ha añadido la ficha y el número de jugador que es.
- messages-resistencia: Mensaje del servidor a todos los clientes de la partida, se envía como respuesta al anterior mensaje. Les envía la casilla en la que añadir la ficha de resistencia, y de qué jugador es la ficha a añadir.
- virus: Mensaje enviado de un jugador al servidor, informando de que ha añadido una ficha infectada por virus al tablero. Le envía la casilla en la que ha añadido la ficha y el número de jugador que es.
- messages-virus: Mensaje del servidor a todos los clientes de la partida, se envía como respuesta al anterior mensaje. Les envía la casilla en la que añadir la ficha infectada por virus, y de qué jugador es la ficha a añadir.
- destroybacteria: Mensaje enviado de un jugador al servidor, informando de que ha destruido una ficha de bacteria normal del tablero. Le envía la casilla en la que ha destruido la ficha, el número de jugador que es y otra variable. Esa variable es de tipo booleano, si es true indica que el jugador ha destruido una bacteria del juego (contará de cara a las estadísticas), y si es false lo que ha hecho es quitar la bacteria del tablero para cambiarla por otro tipo de ficha (resistencia, antibiotico o virus) así que no contará de cara a las estadísticas como bacteria eliminada.
- messages-destroybacteria: Mensaje del servidor a todos los clientes de la partida, se envía como respuesta al anterior mensaje. Les envía la casilla en la que destruir la ficha de

bacteria, de qué jugador es la ficha a destruir y la variable booleana anteriormente explicada.

- destroybiofilm: Mensaje enviado de un jugador al servidor, informando de que ha destruido unas fichas de biofilm del tablero. Le envía la casilla en la que ha destruido las fichas y el número de jugador al que pertenecen las fichas destruidas.
- messages-destroybiofilm: Mensaje del servidor a todos los clientes de la partida, se envía como respuesta al anterior mensaje. Les envía la casilla en la que destruir las fichas de biofilm, y de qué jugador es la ficha a destruir.
- destroyantibiotico: Mensaje enviado de un jugador al servidor, informando de que ha destruido una ficha de antibiotico del tablero. Le envía la casilla en la que ha destruido la ficha y el número de jugador al que pertenece la ficha de antibiotico destruida.
- messages-destroyantibiotico: Mensaje del servidor a todos los clientes de la partida, se envía como respuesta al anterior mensaje. Les envía la casilla en la que destruir la ficha de antibiotico, y de qué jugador es la ficha a destruir.
- destroyresistencia: Mensaje enviado de un jugador al servidor, informando de que ha destruido una ficha de resistencia del tablero. Le envía la casilla en la que ha destruido la ficha y el número de jugador al que pertenece la ficha de resistencia destruida.
- messages-destroyresistencia: Mensaje del servidor a todos los clientes de la partida, se envía como respuesta al anterior mensaje. Les envía la casilla en la que destruir la ficha de resistencia, y de qué jugador es la ficha a destruir.
- destroyvirus: Mensaje enviado de un jugador al servidor, informando de que ha destruido una ficha infectada por virus del tablero. Le envía la casilla en la que ha destruido la ficha, el número de jugador al que pertenece la ficha infectada destruida y una variable booleana, que funciona como en el mensaje de destruir bacterias normales. Si es true, significa que elimina la ficha del juego. Si es false, indica que quita la bacteria infectada por virus del tablero para añadir una bacteria normal (esto sucede cuando el jugador utiliza una carta de antiviral).
- messages-destroyvirus: Mensaje del servidor a todos los clientes de la partida, se envía como respuesta al anterior mensaje. Les envía la casilla en la que destruir la ficha infectada

por virus, de qué jugador es la ficha infectada, y la variable booleana explicada anteriormente.

- robarcarta: Mensaje enviado de un jugador al servidor, indicando que quiere robar una carta de la baraja.
- messages-robarcarta: Mensaje del servidor como respuesta al jugador anterior. El servidor extrae una carta de la baraja asignada a la sala, y le manda esa carta al jugador que se la ha solicitado.
- cartavuelve: Mensaje enviado de un jugador al servidor, en el que el jugador devuelve una carta que acaba de usar a la baraja de la partida. El servidor guarda esa carta al final de la baraja,
- bacteriasdestruidas: Mensaje enviado de un jugador al servidor, se envía cuando un jugador destruye una ficha que contabiliza de cara a las estadísticas. El jugador envía la lista de todas las bacterias que ha destruido de cada jugador rival. Esto sirve para que el servidor guarde ese dato, y en caso de desconexión del jugador, no se pierdan esas estadísticas. Si un jugador se reconecta en el lugar que ha quedado libre, recibirá esos datos para que parta desde el estado de la partida en el que se desconectó el anterior jugador.
- pediobjetivos: Mensaje enviado de un jugador al servidor, ocurre cuando el jugador utiliza la segunda carta de investigador. Esa carta de investigador permite al jugador intercambiar su objetivo con el de otro jugador rival, y este mensaje le sirve al jugador para pedir los objetivos que tienen el resto de jugadores de la partida.
- objetivossala: Mensaje del servidor como respuesta al jugador anterior. En él le envía todos los objetivos que tienen los jugadores de la partida, para que el jugador decida si quiere alguno de ellos o no.
- cambiarobjetivos: Mensaje enviado de un jugador al servidor, sucede si el jugador decide que sí quiere cambiar su objetivo con el de otro jugador. El jugador envía que jugador es él, el objetivo que tiene actualmente y a que jugador quiere cambiarle el objetivo.
- intercambioobjetivo: Mensaje enviado del servidor a un jugador, es un mensaje que se envía después de recibir el anterior. El mensaje se envía al jugador al que se le quería cambiar el objetivo. Los datos que incluye el mensaje son el nuevo objetivo que tendrá este

jugador (objetivo del jugador que solicita el intercambio) y el jugador que ha pedido el cambio, para notificar al destinatario quien ha sido el culpable.

ANEXO IV – LIBRO DE INSTRUCCIONES JUEGO DE MESA

En este apartado se anexará el libro de instrucciones del juego mesa. En ese libro aparecen explicadas todas las reglas del juego y como se deben desarrollar las partidas. Como las reglas del juego online son las mismas que las del juego de mesa, se puede usar como referencia para aprender a jugar y cómo funcionan las partidas del juego. Además, se muestran los elementos gráficos que se usan en este juego online, como son las distintas fichas de bacterias y las distintas cartas que aparecerán en el juego.

Cartas del juego



Generar matriz de biofilm

Añade biofilm alrededor de una de tus bacterias



Gen de producción de antibiótico

Una bacteria produce antibióticos para eliminar bacterias rivales



Gen de resistencia

Una bacteria produce genes de resistencia frente a antibióticos



Quorum sensing

Copia genes de bacterias que estén a 2 o menos casillas de distancia



Antibiofilm

Elimina fichas de biofilm



Cepillo de limpieza

Elimina hasta 3 bacterias adyacentes y su biofilm



Antimicrobiano

Elimina hasta 7 bacterias rivales no protegidas por biofilm



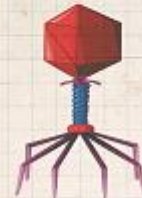
Estropajo sucio

Traslada una bacteria a cualquier parte del tablero



Virus bacteriófago

Infecta bacterias rivales con virus



Antiviral

¡Cura sana, culito de rana...! Elimina la infección de tus bacterias!



Investigador

¡Eureka! Tus nuevos descubrimientos te permiten utilizar las cartas de antimicrobiano frente a bacterias y biofilm



Investigador

Descubre el objetivo de tus rivales





Bacterfield es un juego de mesa de estrategia,
de 2 a 4 jugadores, en el que controlarás el
crecimiento de las bacterias



Bacterfield es un juego de mesa de estrategia, de 2 a 4 jugadores, en el que podrás controlar el crecimiento de las bacterias para conseguir tu objetivo y ganar la partida.

Este juego ha sido creado por Elena Atrián e Isabel Franco, con la financiación del proyecto "Cuenta la Ciencia" de la Fundación General CSIC. El objetivo de este proyecto de divulgación es dar a conocer el apasionante mundo de las bacterias y... ¡qué mejor forma que aprender jugando!

Con Bacterfield descubrirás que estos pequeños microorganismos son capaces de hacer muchas más cosas de las que pensamos, como producir antibióticos o comunicarse entre ellas.

Reúne tus bacterias, planifica tu estrategia, utiliza las cartas para adquirir nuevos genes o para eliminar a las bacterias rivales y.... ¡que comience la partida!

Edad recomendada: a partir de 14 años.

Contenido del juego

71 cartas que se dividen en:

4 tipos de cartas especiales (6 de cada tipo = 24)

6 tipos de cartas de acción (6 de cada tipo = 36)

2 cartas de investigador

5 cartas de objetivo

4 cartas resumen

*El juego viene con un pack adicional de 71 cartas en inglés.

600 fichas de bacterias:

150 fichas azules (50 de cada tipo)

150 fichas naranjas (50 de cada tipo)

150 fichas moradas (50 de cada tipo)

150 fichas verdes (50 de cada tipo)

200 fichas de biofilm:

50 fichas azules

50 fichas naranjas

50 fichas moradas

50 fichas verdes

1 tablero

Preparación

Reparte todos los componentes necesarios a cada jugador:

- 150 fichas de bacterias

- 50 fichas de biofilm

- 1 carta de objetivo

- 1 carta de juego

coloca el tablero y el mazo de cartas de forma que esté accesible para todos los jugadores.

coloca una ficha de bacteria normal en la casilla de salida de cada jugador.

Objetivo del juego

cumple tu objetivo antes que el resto de jugadores para ganar la partida. La carta de objetivo que se ha repartido al inicio te indicará qué objetivo tienes que cumplir. ¡Para ello es muy importante que planees una estrategia desde el principio y que utilices sabiamente las cartas de juego!

cada jugador tendrá un objetivo distinto con el que conseguirá ganar la partida. El objetivo se reparte al inicio, durante la preparación del juego, y será privado para cada jugador. El juego dispone de 5 objetivos distintos, de esta forma siempre habrá un objetivo que no se utilizará y así será más difícil descubrir el objetivo de tus rivales!

Objetivos individuales

Conquistar 48 casillas: para ganar la partida tendrás que conquistar 48 casillas del tablero, ¡todas las casillas cuentan!

Llegar a la casilla de salida de un jugador rival: consigue colonizar la casilla de salida de cualquier jugador rival para ganar la partida. Si hay menos de 4 jugadores no se podrán utilizar las casillas de salida vacías.

Colocar 6 bacterias en casillas bacteriostáticas: gana la partida colonizando 6 casillas bacteriostáticas antes de que el resto de jugadores alcancen su objetivo.

Eliminar 24 bacterias rivales: ¡se el primero en alcanzar tu objetivo utilizando sabiamente las cartas de juego para eliminar a tus rivales! Ganarás una vez hayas conseguido eliminar 24 bacterias en total, sin importar el número de bacterias eliminadas de cada jugador.

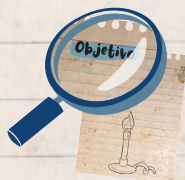
Modificación para 3 jugadores: Eliminar 22 bacterias

Modificación para 2 jugadores: Eliminar 20 bacterias

Eliminar 8 bacterias de cada jugador rival: al igual que en el objetivo anterior tendrás que ayudarte de las cartas de juego para conseguir eliminar 8 bacterias de cada jugador rival y así ser el primero en alcanzar tu objetivo y ganar la partida.

No incluir este objetivo cuando haya menos de 4 jugadores.

¡Recuerda que sólo podrás ganar la partida si consigues tu objetivo con bacterias no infectadas por virus!



Desarrollo del juego

Cada jugador comienza la partida con una bacteria normal en la casilla de salida y una carta de juego. El turno se desarrolla de la siguiente manera:

FASE 1 - Producir antibiótico

Conforme avanza la partida las bacterias pueden ir adquiriendo distintos genes que te ayudarán a cumplir con tu objetivo, así que lo primero a tener en cuenta al inicio del turno será comprobar si alguna de tus bacterias produce antibióticos. En caso afirmativo elimina todas las bacterias rivales que se encuentren en contacto directo con la bacteria productora de antibiótico, siempre que estas no estén protegidas por biofilm o posean un gen de resistencia.

FASE 2 - Elegir acción

1) Utilizar una carta especial y expandir tu bacterias:

Añade una bacteria por cada bacteria situada en el tablero que tenga casillas contiguas libres y que no esté situada en una casilla bacteriostática. Esta acción te permite utilizar también una carta especial antes de expandir tus bacterias por el tablero.

2) Utilizar una carta especial y una carta de acción:

Si no te interesa expandir tus bacterias, ¡esta es tu opción! Puedes utilizar una carta especial y después una carta de acción.

FASE 3 - Robar

Roba una carta del mazo. Se pueden tener un máximo de 5 cartas en la mano, así que si ya tienes 5 cartas descarta una antes de robar.

¡Es obligatorio robar! No puedes quedarte con las 5 cartas y pasar el turno sin haber robado.

Si robas una carta de investigador tendrás que utilizarla al principio de tu siguiente turno, de esta manera no afectará a las fases del turno.

Tablero y tipos de casillas

El tablero está formado por 180 casillas, que se dividen en:

Casillas de salida

Son las casillas de colores situadas en la zona central del tablero. Al inicio de la partida se colocará una bacteria de cada jugador en su casilla de salida correspondiente.

*No se puede utilizar la carta "Estropajo sucio" sobre la casilla de salida.



Casillas de crecimiento



Estas casillas permiten el avance de las bacterias por el tablero. Coloca las fichas de bacterias en el centro de la casilla y las fichas de biofilm en los vértices.

Casillas bacteriostáticas



Ayudan a controlar el crecimiento bacteriano impidiendo la reproducción de la bacteria. Las bacterias que se sitúen en estas casillas no darán lugar a otra bacteria cuando se dividan, pero pueden adquirir genes y producir biofilm.

Tipos de fichas

Bacterias



Estas fichas te permiten expandirte por el tablero. En cada ronda podrás colocar una bacteria, en un hexágono adyacente y libre, por cada bacteria que haya en el tablero. Ejemplo: si tienes 1 bacteria, se duplica y al final del turno tendrás 2, si tienes 2 se duplicará cada una y al final del turno tendrás 4. Recuerda que las bacterias situadas en casillas bacteriostáticas no se dividen. Si durante la partida eliminan todas tus bacterias, empieza el siguiente turno colocando una bacteria normal en la casilla de salida.

Producción de antibiótico



Se utiliza con la carta "Gen de producción de antibiótico" y se aplica sustituyendo la ficha de bacteria que elijas por esta ficha. Estas bacterias pueden eliminar bacterias no protegidas por biofilm ni con gen de resistencia. Todas las bacterias que se dividan a partir de esta tendrán también el gen de producción de antibiótico.

Resistencia a antibióticos



Se utiliza con la carta "Gen de resistencia" y se aplica sustituyendo la ficha de bacteria que elijas por esta ficha. Estas bacterias son resistentes a los antibióticos producidos por bacterias rivales. Todas las bacterias que se dividan a partir de esta tendrán también el gen de resistencia.

Bacterias infectadas



Esta ficha se encuentra en el reverso de todas las fichas anteriores y se utiliza cuando un rival infecta tus bacterias con la carta "Virus bacteriófago". Las bacterias infectadas por virus no pueden utilizar ni adquirir genes, ni producir biofilm. Las bacterias infectadas se replican con el virus, es decir, todas las fichas que se dividan a partir de una bacteria infectada darán lugar a una bacteria infectada y sin genes de producción o de resistencia a antibióticos.

Biofilm



Podrás usar las cartas "Generar matriz de biofilm" para colocar 6 fichas de biofilm en los vértices de una casilla, alrededor de la bacteria que tú elijas. El biofilm protege a las bacterias de los antimicrobianos, los antibióticos y los virus. Una bacteria estará protegida si hay, al menos, una ficha de biofilm en cualquiera de los vértices de su casilla, sin importar el color de la ficha de biofilm. Es decir, el biofilm de un jugador también protege las bacterias rivales. Las fichas de biofilm se pueden colocar en cualquier tipo de casilla.

Cartas de juego

Cartas especiales

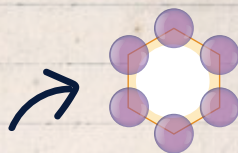
Estas cartas te permitirán aprovechar diferentes acciones que pueden realizar las bacterias por sí mismas.



Generar matriz de biofilm

Elige una bacteria y rodéala colocando 6 fichas de biofilm en los vértices de la casilla.

El biofilm protege a todas las bacterias frente a antibióticos, antimicrobianos y virus.



Gen de producción de antibiótico

Esta carta te permite convertir una bacteria normal en una bacteria productora de antibióticos. El antibiótico tendrá efecto a partir del siguiente turno y eliminará todas las bacterias rivales situadas en casillas adyacentes siempre que no estén protegidas por biofilm o tengan genes de resistencia. Las bacterias eliminadas se retirarán del tablero y se las quedará el jugador que las haya eliminado.



Gen de resistencia

Esta carta te permite convertir una bacteria normal en una bacteria resistente a antibióticos. Las bacterias con gen de resistencia no podrán ser atacadas por antibióticos, pero sí por antimicrobianos y virus.



Quorum sensing

Las bacterias pueden comunicarse entre ellas gracias al quorum sensing, lo que te permitirá copiar un gen de una bacteria (tuya o de cualquier jugador) que esté a 2 o menos casillas de distancia.



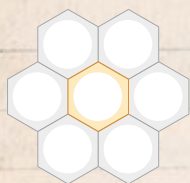
Cartas de acción

Estas cartas te ayudarán a luchar contra las bacterias rivales para cumplir tu objetivo.

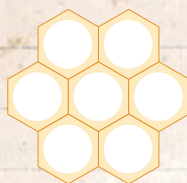
Antimicrobiano

Elige una bacteria y elimina esa y todas las bacterias contiguas, hasta un máximo de 7 bacterias rivales.

¡Ten en cuenta que este antimicrobiano es de amplio espectro y también se eliminan tus propias bacterias si entran en el radio de acción! Las bacterias protegidas por biofilm no se eliminan.



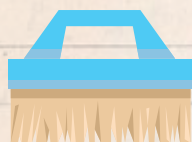
1) Elige la casilla donde quieras usar el antimicrobiano.



2) Elimina esa bacteria y todas las que estén a su alrededor.

Antibiofilm

Elimina todas las fichas de biofilm de la bacteria que elijas.



Cepillo de limpieza

Elimina hasta 3 bacterias adyacentes y todo el biofilm que las rodea, sean del color que sean. Las bacterias pueden estar en las siguientes posiciones:



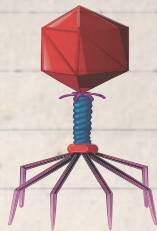
Estropajo sucio

Traslada una bacteria a cualquier casilla, excepto a las casillas de salida. Si la casilla está ocupada, la bacteria que la ocupa es eliminada, aunque esté protegida por biofilm.



Virus bacteriófago

Infecta una bacteria rival y todas las que estén a su alrededor, hasta un máximo de 7 bacterias rivales. Las bacterias infectadas se replicarán con el virus, es decir, cada bacteria infectada dará lugar a una nueva bacteria infectada. Las bacterias infectadas no pueden generar biofilm ni utilizar, adquirir o transmitir genes. Además, no contabilizarán a la hora de cumplir el objetivo. Las bacterias con biofilm no pueden ser atacadas por virus. Si una bacteria infectada recibe un nuevo virus es eliminada.



1) Elige la bacteria que quieras infectar.



2) Dale la vuelta a esa bacteria y a todas las que estén a su alrededor.



Antiviral

Elimina la infección de todas tus bacterias infectadas por virus.

Cartas de investigador

Las cartas de investigador se utilizan al inicio del siguiente turno, antes de la fase 1, y te proporcionarán una gran ventaja.

Investigador 1



¡Si consigues convertirte en este investigador podrás utilizar la carta de antimicrobiano también frente a bacterias protegidas por biofilm! Cuando utilices las cartas de antimicrobiano eliminarás las bacterias y todo el biofilm que tengan.

Investigador 2



Este investigador te permite conocer el objetivo de tus rivales. Cuando saques esta carta los demás jugadores deberán enseñarte sus cartas de objetivo, y tú podrás decidir si quieres cambiar el tuyo por otro.

Fin de la partida

La partida finalizará cuando un jugador logre cumplir su objetivo.

Modo de juego adicional

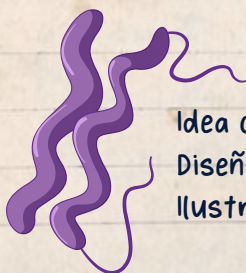
Se puede jugar un modo adicional sin objetivos individuales. En este modo el objetivo común será colonizar la mayor parte de tablero, y ganará el jugador que consiga conquistar antes 36 casillas.

Agradecimientos

El juego ha sido desarrollado gracias a la financiación de la III edición del programa "cuenta la ciencia", de la Fundación General CSIC, y con el apoyo institucional del Instituto de Nanociencia y Materiales de Aragón (INMA), Consejo Superior de Investigaciones Científicas (CSIC), Universidad de Zaragoza, Centro de Investigación Biomédica en Red (CIBER-BBN), Comisión Europea, MSCA-IF-2018 (Grant agreement number: 845427) y Gobierno de Aragón.

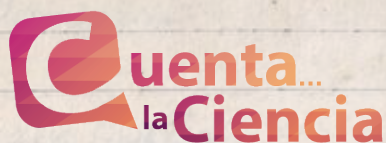
Queremos agradecer a nuestros compañeros (y amigos) de trabajo, que nos han ayudado muchísimo con el desarrollo del juego.

Nos gustaría agradecer también a Raquel García por las preciosas ilustraciones que ha realizado para el diseño final de Bacterfield.



Idea original y derechos intelectuales: Elena Atrián e Isabel Franco
Diseño del juego: Elena Atrián e Isabel Franco
Ilustraciones: Raquel García

Con la financiación de:



Con el apoyo institucional de:



**Universidad
Zaragoza**

ciber-bbn







**European
Commission**

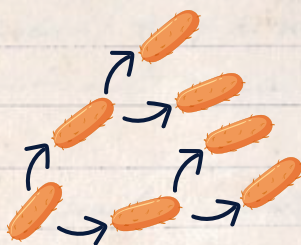
Horizon 2020
European Union funding
for Research & Innovation



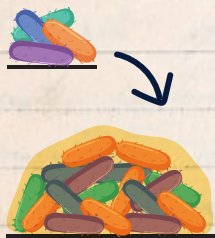
¿Sabías que...?

Las bacterias son microorganismos unicelulares y son los protagonistas de este juego. Pueden tener formas muy distintas: bacilos , cocos , espirilos  o vibrios  y... ¡algunas son beneficiosas para nosotros! Por ejemplo, las bacterias que se encuentran en el intestino y nos ayudan a digerir la comida. Pero también hay bacterias que pueden causar enfermedades tan graves como la peste (causada por la bacteria *Yersinia pestis*), salmonelosis (*Salmonella* spp.) o cólera (*Vibrio cholerae*).

La ventaja que tienen las bacterias es que crecen muy rápido, una bacteria es capaz de generar otra bacteria a partir de ella misma en muy poco tiempo. ¡Algunas, como *Escherichia coli*, tardan tan solo 20 minutos en duplicarse!



Además, este tipo de crecimiento es exponencial, lo que quiere decir que cada vez que las bacterias crecen se duplica su población. Pero tampoco pueden crecer de forma infinita: conforme crecen van agotando los nutrientes y llegan al estado de "fase estacionaria", donde permanecen metabólicamente activas, pero no se dividen más.



Las bacterias pueden formar sus propios ecosistemas en distintas superficies, llamados biofilm (o biopelícula), en los que además se protegen de agentes externos gracias a la liberación de distintas sustancias que constituyen una matriz protectora.

Esta matriz está formada principalmente por polisacáridos (como azúcares), ADN extracelular y proteínas y no solo les otorga protección, sino que además les facilita la comunicación entre ellas. Esta forma de comunicación celular, conocida como "quorum sensing", les permite regular la expresión de distintos genes para adaptarse al medio.

¡Pero es que además pueden intercambiarse genes entre ellas! Las bacterias poseen un tipo de material genético conocido como plásmidos que les pueden otorgar nuevas características, como la temida resistencia a antibióticos. Pero más fascinante aún es como lo hacen: pueden pasarse estos plásmidos entre ellas gracias a un proceso de conjugación por el que dos bacterias se comunican mediante unos "pelos" (pili), creando un conducto entre ellas.



Como puedes ver, las bacterias son seres más complejos de lo que se podría pensar a pesar de ser microorganismos unicelulares.

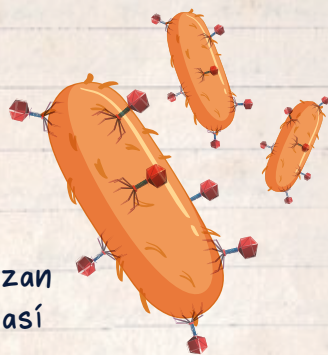
Para combatir las infecciones bacterianas disponemos de muchos tipos distintos de antimicrobianos. Estos agentes pueden actuar de dos maneras: matando a las bacterias mediante un efecto bactericida o simplemente inhibiendo su crecimiento mediante un efecto bacteriostático. Hay antimicrobianos de muchos tipos, como la lejía, que se utiliza para desinfectar superficies, o los antibióticos, que se utilizan para tratar infecciones que ocurren dentro del cuerpo. ¿Pero sabías que muchos antibióticos son producidos por los propios microorganismos? Aunque hoy en día se sintetizan en el laboratorio, los microorganismos llevan mucho tiempo produciendo estos agentes para luchar contra otros microbios. De hecho, el primer antibiótico descubierto fue la penicilina, que se llama así en honor a un hongo del género *Penicillium*, y fue fruto de un accidente de laboratorio, en el que el hongo contaminó una placa inoculada con bacterias e impidió el crecimiento de estas a su alrededor.

Pero... ¿qué pasa si las bacterias que estamos intentando eliminar están protegidas por biofilm? Pues que probablemente nuestro antimicrobiano no logre llegar a ellas ya que el biofilm actúa como una "coraza" que las protege de los agentes externos. Para ello necesitamos crear antimicrobianos específicos con actividad antibiofilm, que permita destruirlo o que inhiba su formación y así dejar las bacterias accesibles al antimicrobiano. Aquí entran en juego la ciencia y los investigadores, que trabajan cada día para desarrollar nuevos compuestos con mejores propiedades.

La vida de las bacterias no es nada fácil, ¡incluso pueden enfermar como nosotros! Hay virus específicos, los virus bacteriófagos, que pueden infectar a las bacterias y causar su muerte.

Funcionan de manera análoga a los virus que nos enferman a nosotros: entran en una célula y utilizan su maquinaria celular para producir más virus y así ir infectando a más y más células.

Es esta dependencia a otras células lo que los convierte en parásitos intracelulares obligados y por lo que aún existe el debate de si son seres vivos o no, ya que no pueden reproducirse por si mismos y siempre necesitan una célula huésped donde completar su ciclo. Por lo tanto, para eliminarlos no podemos utilizar antibióticos, ya que estos están dirigidos a estructuras propias de las bacterias de las que los virus carecen, y necesitamos unos agentes específicos, los antivirales, que nos permitan eliminarlos.



¡Ya ves que el día a día de las bacterias no es nada aburrido! con *Bacterfield* podrás adentrarte en su fascinante mundo y descubrir muchas de las cosas que estos pequeños microorganismos son capaces de hacer, porque... ¡qué mejor que aprender jugando!